



Götz, Dennis

Titel der Abschlussarbeit - Die Weiterentwicklung des Browsers
dank der Implementierung von HTML5/CSS3D sowie die Funktion
des Echtzeit-Renderings und die damit verbundene Erschaffung ei-
nes hardware-unabhängigen Mediums.

- Bachelorarbeit -

Hochschule Mittweida - University of Applied Science (FH)

Berlin - 2010



Götz, Dennis

Titel der Abschlussarbeit - Die Weiterentwicklung des Browsers dank der Implementierung von HTML5/CSS3D sowie die Funktion des Echtzeit-Renderings und die damit verbundene Erschaffung eines hardware-unabhängigen Mediums.

- eingereicht als Bachelorarbeit -

Hochschule Mittweida - University of Applied Science (FH)

Erstprüfer
Prof. Dr. Peter Will

Zweitprüfer
Dipl. Ing. Andreas Hog

Berlin - 2010

Bibliographische Beschreibung

Götz, Dennis

Die Weiterentwicklung des Browsers dank der Implementierung von HTML5/CSS3D sowie die Funktion des Echtzeit-Renderings und die damit verbundene Erschaffung eines hardware-unabhängigen Mediums. - 2010 - 60 Seiten.

Berlin, Hochschule Mittweida, Fachbereich Medien, Bachelorarbeit

Referat

Diese Bachelorarbeit untersucht, ob die neuentwickelten Verfahren und Technologien, den bisherigen Browser in ein eigenständiges Medium umwandeln können. Neue Möglichkeiten und Features erweitern das Internet und zugleich auch den dazu benötigten Browser. Derzeit schon benutzte Praktiken, sollen dazu beitragen die These zu stützen.

Inhaltsverzeichnis

Abbildungsverzeichnis	... S. 6
Abkürzungsverzeichnis	... S. 8
1. Einleitung	
1.1 Erläuterung	... S. 9
2. Hauptteil	
2.1 Geschichte und Entstehung des Browsers	... S. 10
2.2 Browserweichen	... S. 14
2.3 Funktionen und Möglichkeiten	
2.3.1 HTML5	... S. 15
Besonderheiten	... S. 18
Offline-Anwendungen	... S. 20
Multimedia Elemente und eingebettete Inhalte	... S. 22
Drag & Drop API	... S. 23
Das Canvas Element	... S. 25
2D-Bereich	... S. 25
3D-Bereich	... S. 27
Zwischenfazit	... S. 29
Zukunft von HTML5	... S. 29
2.3.2 WebGL	... S. 30
2.3.3 CSS Level 3	... S. 31
Um die Z-Achse erweitert	... S. 34
Verdrängung von Flash?	... S. 36
2.3.4 Hardwareunabhängiges Medium...aber wie?	... S. 38

Inhaltsverzeichnis

2.3.5 Streaming Media	... S. 38
QuakeLive	... S. 39
OnLive	... S. 40
Vorteile des Streamens	... S. 42
Nachteile des Streamens	... S. 43
2.3.6 Cloud Computing	... S. 44
2.3.7 Google Chrome OS	... S. 46
2.4. Einmal bitte Updaten...	... S. 49
3. Schlussteil	
3.1 Fazit	... S. 51
3.2 Standpunkt in 5 bis 10 Jahren	... S. 52
4. Literaturverzeichnis	... S. 54
5. Erklärung zur selbstständigen Anfertigung	... S. 59

Abbildungsverzeichnis

Abb. 1 - Screenshot des NSCA Mosaic Browser 1.0	... S. 10
Abb. 2 - Marktanteile verschiedener Browser von Januar 2002 bis Mai 2010	... S. 12
Abb. 3 - Screenshot von der Spiegel Online Webseite aus dem Jahre 1998	... S. 15
Abb. 4 - Screenshot der heutigen spiegel.de Webseite	... S. 16
Abb. 5 - Screenshot der Downloadstatistik von Firefox	... S. 25
Abb. 6 - Screenshot des Leveleditors von CopperLicht	... S. 27
Abb. 7 - Screenshot des CopperLicht Projekte: Asteroid Game	... S. 28
Abb. 8 - Livedemo von Google's O3D Schnittstelle	... S. 31
Abb. 9 - CSS3 Slideanimation im Menübeispiel	... S. 33
Abb. 10 - 3D-Würfel-in-Würfel mit CSS generiert	... S. 34
Abb. 11 - Screenshot von Cooliris	... S. 35
Abb. 12 - Screenshot von Quake Live im Browser	... S. 39
Abb. 13 - OnLive - How it works	... S. 41
Abb. 14 - OnLive Auswahlmenü von mehreren Spieltitel	... S. 43
Abb. 15 - Darstellung des Cloud Computing Prinzips / IT-Struktur	... S. 45
Abb. 16 - Umfrage zu dem Betriebssystem Chrome OS von chip.de	... S. 47

Abb. 17 - Mögliches Design von Chrome OS	... S. 48
Abb. 18 - Statistik der Marktanteile des Browsers in Deutschland	... S. 48
Abb. 19 - Statistik der weltweiten Browser-Marktanteile	... S. 49
Abb. 20 - Verlauf der Marktanteile von Januar - Juli 2010	... S. 51

Abkürzungsverzeichnis

HTML	Hypertext Markup Language / Hypertext-Auszeichnungssprache
CSS3	Cascading Style Sheets Level 3
API	Application programming interface z.B. Plugin/eigenständiges Interface
O3D	Google's opensource JavaScript API
WebGL	Webstandard zur Implementierung von 3D API's in Browsers
GUI	Graphic user interface / Benutzerinterface
open source	Quelltext wird veröffentlicht und kann weiter entwickelt werden
FPS	Frames per second / Bilder pro Sekunde
Tag	HTML-Tag z.B. <p>, <body>

1.1 Erläuterung

Innerhalb der letzten paar Jahre haben sich einige neue Technologien und Trends entwickelt die das Internet revolutionierten. In dieser Arbeit möchte ich anhand von einigen Beispielen zeigen und erläutern, welche neuen Möglichkeiten das Internet hergibt, die mich erst auf die Idee meiner oben genannten These brachten.

Ein Hauptteil der Arbeit wird sein, ihnen durch die Implementierung von HTML5 und CSS3D im Browser, die neuen Funktionen und Möglichkeiten darzustellen und anhand von praktischen Beispielen die Auswirkungen zu erklären. Der andere Teil der Arbeit fokussiert sich auf die Weiterentwicklung von der Streaming - und Cloud Computing Technologie im Bereich der hardwarelastigen 3D-Anwendungen (Computerspiele).

Ich selber bin seit Jahren begeisterter Fan von Computerspielen. Umso grafisch anspruchsvoller und realistischer, desto besser. Damit man aber immer die aktuellsten Computerspiele einwandfrei und flüssig auf seinen Rechner zu Hause nutzen kann, sollte man seine Hardware auf einen relativ neuen Stand halten. So ist es in der Regel notwendig, alle zwei bis maximal vier Jahre, seinen PC aufzurüsten bzw. komplett neu zu kaufen, da er sonst nur noch zu Anwendungen im Officebereich dient. Die neu produzierten Titel werden grafisch immer detaillierter und ressourcenlastiger, welche die gesamte Hardware mit Unmengen an Berechnungen sehr belastet. Die Hauptlast liegt hier bei der Grafikkarte, dem Prozessor und dem Arbeitsspeicher. Eine Aufrüstung mit neueren Komponenten hat zwar hinterher den wiedergewonnenen Spielspaß zur Folge, aber auch meist ein leeres Portemonnaie.

Einige Firmen und Entwickler haben dieses Problem erkannt und sich dazu Lösungen einfallen lassen. Bisher gibt es dutzende erfolgreicher Prototypen und Plattformen, die anhand der Weiterentwicklung der Streaming Technologie, den PC des Benutzers deutlich entlasten und lediglich eine schnelle Internetverbindung voraussetzen.

Zu guter Letzt, nach Betrachten aller Fakten und geplanten Erneuerungen oder technologischen Möglichkeiten, werde ich ein Fazit ziehen und eine mögliche Prognose erstellen, wo wir eventuell in den nächsten 5-10 Jahren stehen könnten.

2.1 Geschichte und Entstehung des Browsers

Das Internet, so wie wir es heutzutage kennen, wurde damals von Tim Berners-Lee¹ erfunden. Er hatte im Jahr 1989 am europäischen Kernforschungszentrum CERN die Idee, mit Hilfe von Hypertext, den Informationsaustausch zwischen Wissenschaftlern zu vereinfachen, indem gesammelte Daten auf einem Server hoch geladen und aktualisiert werden konnten. Er entwickelte das Hypertext Transfer Protocol (HTTP), dass als Standard für die Kommunikation zwischen dem Client und dem Server gelten sollte. Damit der Benutzer die Internetseiten auf dem Server leicht erreichen kann, erfand er den Universal Resource Identifier (URL), welcher ihm gestattet eine Ressource auf dem Webserver zu finden und aufrufen. Die zuletzt eingeführte Auszeichnungssprache Hypertext Markup Language (HTML) dient bis heute noch als Standard zum Aufbau und der Strukturierung von Internetseiten.

Die Darstellung der Daten erfolgte mit dem ersten Browser namens *WorldWideWeb*, der in der Lage war den HTML-Code auszulesen und eine grafische Oberfläche erzeugte. In den folgenden Monaten und Jahren wurden immer mehr Entwickler und Studenten an Universitäten auf diese Erfindung aufmerksam und entwickelten ihre eigenen Browser.

Am 1. April 1993 wurde der an dem *National Center for Supercomputing Applications* entwickelte Browser "NCSA Mosaic 1.0" veröffentlicht. Er enthielt eine übersichtliche GUI mit allen nötigen Werkzeugen um benutzerfreundliches navigieren auf Internetseiten zu gewährleisten. Des Weiteren war Mosaic 1.0 in der Lage, Texte und Grafiken ohne vorherigens verlinken, selbstständig zu laden.

¹ http://de.wikipedia.org/wiki/Tim_Berners-Lee#Berners-Lee_und_das_World_Wide_Web

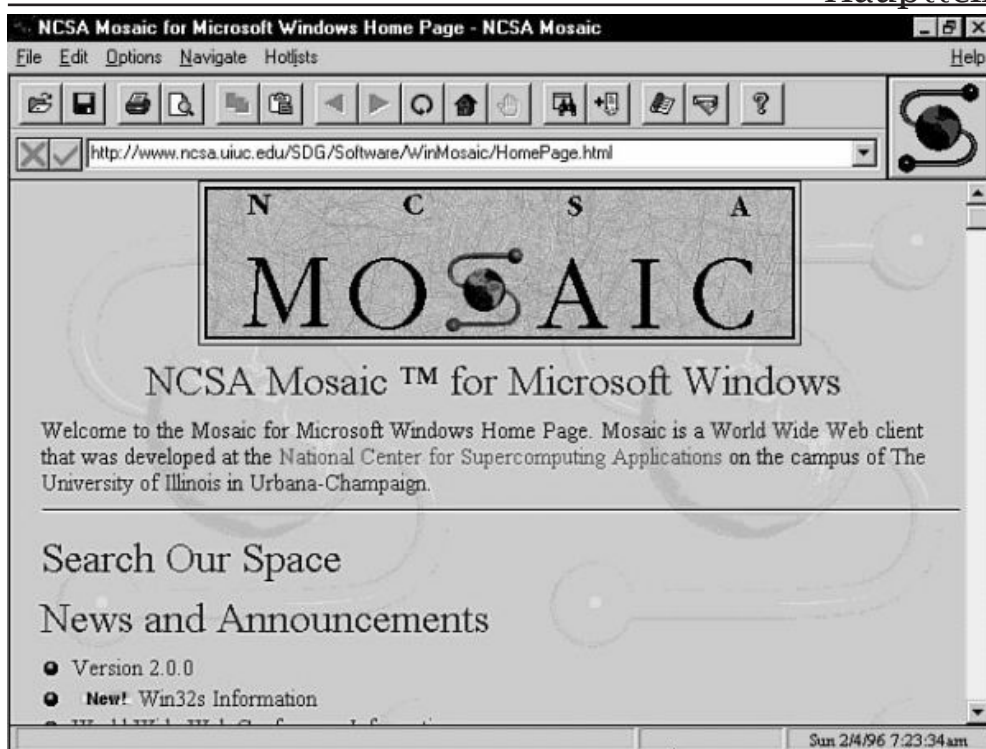


Abb. 1 - Screenshot des NSCA Mosaic Browser 1.0

Die kurze Vorherrschaft gegenüber anderen Browsers wurde schnell beendet, nachdem die Firma *Netscape Communications Corporation* gegründet wurde und den Netscape Navigator veröffentlichte (Marktanteil über 80%)². Erst ab der Netscape Version 4.x.x wurde der Programmcode zur open source, daraus folgte dann das Mozilla-Projekt. Die folgenden Versionen von Netscape hatten keinen großen Erfolg mehr und der Marktanteil verringerte sich deutlich. Grund dafür war der Microsoft Internet Explorer (IE), der beim Betriebssystem schon mitgeliefert wurde. In den folgenden Jahren war der Microsoft Internet Explorer der unangefochtene Browser und hatte keine Konkurrenz zu befürchten. Er war ein fester Bestandteil in Microsoft's Betriebssystemen ab Windows 95 und wurde ständig weiter entwickelt. 1998 wurde das Unternehmen Netscape Communications Corporation schließlich von *America Online* (AOL) aufgekauft, in der vergeblichen Hoffnung, den Browserkrieg doch noch zu gewinnen.

Am 23. September 2002 brachte die Mozilla Foundation ihren

² http://de.wikipedia.org/wiki/Netscape_Navigator

Hauptteil

ersten freien Webbrowser *Phoenix* auf dem Markt, der auf der Gecko Rendering-Engine³ basierte. Das bis dahin von Microsoft verpönte Feature „Tabbed Browsing“, welches erstmals 1994 in dem Browser „Internet Works“ erschien, erlaubte dem Benutzer von Mozilla Phoenix, mehrere Internetseiten in einem Fenster zu öffnen. Die folgenden Versionen *Firebird* und letztlich *Firefox* boten die Möglichkeit, den Browser mit installierbaren Erweiterungen auszustatten, die nicht vom ursprünglichen Browser angeboten wurden. Somit kann man seinen Browser individuell mit Themes⁴ gestalten. Webnavigation mit der Maus steuern oder dank der Adblock Erweiterung ungebetene Flash-Werbung (Popups) oder Inhalte blocken.

Mittlerweile gibt es für Firefox über 150.000 Plugins⁵. Auf der nächsten Seite zeigt eine Statistik die Marktanteile der verschiedenen Browsern vom Jahr 2002 bis Mitte 2010. Hier fallen drei weitere Browser auf - Opera, Safari und Google Chrome.

Opera wurde von dem norwegischen Telekommunikationsunternehmen Telenor als ursprüngliches Forschungsprojekt entwickelt. Erst mit Version 2.0 wurde Opera veröffentlicht und war kostenpflichtig erhältlich. Später konnte man die Browserversion 5-8 kostenlos erwerben, es wurde jedoch ein Werbebanner (Adware) eingebunden. Erst ab dem Jahr 2006 mit Version 8.0 wurde der Browser kostenlos und war ohne Adware erhältlich. Wie man an der Grafik sieht hat Opera nie den großen Durchbruch geschafft, obwohl er der Konkurrenz mit einigen interessanten Funktionen ab Version 4.0 voraus war (z.B. tabbed browsing, integrierter ICQ-Instant Messenger und E-Mail Client). Gründe für den niedrigen Marktanteil, war zum Beispiel der schlechte Ruf von Opera aus den vergangenen Versionen, die unter Instabilität und Programmfehler litten. Ab dem Jahr 2005 hat die Firma Opera Software sich auf eine weitere Sparte konzentriert - Mobiletelefone. Opera Mini bietet für Handynutzer einen Browser für Java-fähige Mobiltelefone und PDAs. Opera Mobile wurde für Pocket-PCs und Smartphones mit Windows Mobile, Symbian OS oder Linux entwickelt. Im Juli 2006 erschien in Kooperation mit Nintendo der Nintendo-DS Browser der kostenpflichtig für Nintendo DS (Lite) Benutzern gängige

³ Software Modul zum Darstellen von Webseiten

⁴ erlauben verschiedene grafische Anpassungen

⁵ <https://addons.mozilla.org/de/firefox/?browse=featured>

Hauptteil

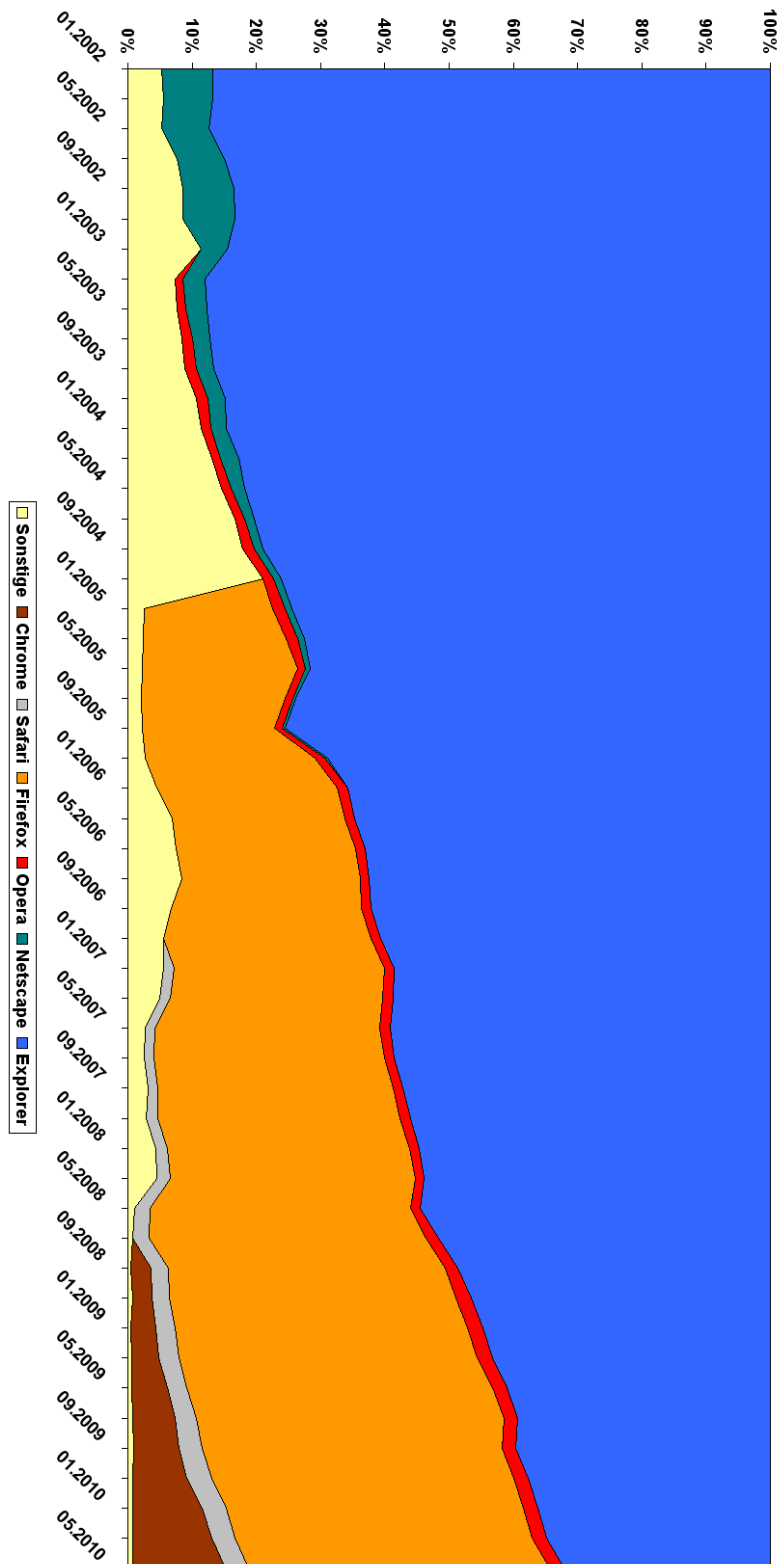


Abb. 2 - Marktanteile verschiedener Browser von Januar 2002 bis Mai 2010

Hauptteil

Webseitendarstellen kann. Weiterhin wurde der Opera-Browser in der Spielkonsole Wii und Nintendo DSi integriert.

Safari ist der als standard-mitgelieferte Webbrowser der Marke Apple für das hauseigene Betriebssystem Mac OS X. Safari hat in jedem web-fähigen Apple Produkt seinen festen Platz (Mac-Book, iPhone, iTunes, iPad etc.). Der Browser ist zwar seit dem 11. Juni 2007 auch für Windows-Benutzer verfügbar, aber findet hier kaum einen Platz.

Google Chrome wurde am 2. September 2008 von Google Inc. veröffentlicht und ist für Windows, Mac OS X und Linux Betriebssystemen erhältlich. Die Besonderheiten bei Chrome sind die einfache Bedienung und die Geschwindigkeit in der sich die Webseiten angezeigt werden. Zu den neusten Sicherheitsfunktionen, lässt sich Google Chrome ebenfalls wie Mozilla Firefox mit Addons aller Art erweitern. Chrome soll hingegen keine Konkurrenz zu Microsofts Internet Explorer darstellen, sondern eher ein weiterer Schritt in Richtung „cloud computing“, auf dessen Technologie Google's Betriebssystem - Google Chrome OS basiert (Kapitel 2.3.7).

2.2 Browserweichen

Durch die damalige rapide Entwicklung von HTML und vor allem CSS, gab es relativ wenige Elemente die das W3C als Standard deklariert hat. So kam, und kommt es heute immer noch vor, dass die CSS Interpretation von unterschiedlichen Browsern oder Versionen der eigene Marke abweicht. Es ist keine Seltenheit, dass sich eine und die selbe Webseite, im einen Browser anders verhält als im Nächsten. Meistens sind einige Designelemente verrutscht oder werden völlig falsch dargestellt.

Das größte Problem stellt der Internet Explorer da. Seit jeher tanzt der Internet Explorer aus der Reihe und hat sich deswegen zu großer Unbeliebtheit bei den Webentwicklern verschafft. Das größte Sorgenkind der IE-Reihe war die Version 6.0, der mit seinen Quirks-Modus weit von dem standardisierten Box-

Hauptteil

Modell abweicht (Positionierungs-Elemente werden völlig falsch interpretiert). Die anderen nennenswerten Browsermarken wie Firefox, Safari, Opera und Chrome interpretieren CSS ziemlich gleich. Man kann davon ausgehen das eine Seite im Firefox genauso aussieht wie in Chrome.

Eine bisher ausreichende Lösung dafür, sind die sogenannten „Conditional Comments“. Sie werden eingesetzt um einen alternativen Code zum Einsatz zu bringen, wenn es sich um die entsprechende Browserversion des Betrachters handelt. Conditional Comments werden wie normale HTML Kommentare an die passenden Stellen im Code implementiert.

```
<!--[if IE]>  
    <style type="text/css">@import url(ie.css);</style>  
<![endif]-->
```

In dem Beispiel wurde extra eine CSS Datei für den Internet Explorer ausgelagert, und kommt nur zum Einsatz sofern der Betrachter mit einem IE die Seite öffnet. Alternative Ansätze wie z.B. „CSS-Hacks“ gibt es ebenfalls, davon wird jedoch von abgeraten. Sie gelten allgemein als „unsauber codiert“ und können zu Problemen mit neueren Browserversionen führen.

2.3 Funktionen und Möglichkeiten

2.3.1 HTML5

Was genau ist nun eigentlich HTML5? Wenn man es genau betrachtet ist es eine Weiterentwicklung von HTML4.01. - welche laut Spezifikation des World Wide Web Consortium⁶ (W3C) als Webstandard definiert wurde, und bis heute noch gilt. Müsste man HTML5 genauer definieren, so ist es nicht in erster Linie die Sprache an sich, sondern das was der Browser damit umsetzt - das Document Model Object (DOM), also Elemente und Schnittstellen einer Baumstruktur. Diese Elemente und Schnittstellen für HTML5 laufen unter den passenden Namen DOM5 und ist das eigentliche Herz des Ganzen.

⁶ www.w3c.org

Hauptteil

Nun wurde HTML 4.01. am 24. Dezember 1999 veröffentlicht, wozu sollte man sich dann die Mühe machen eine neue komplett überarbeitete Version zu entwickeln wenn sich das Ganze schon seit über zehn Jahren bewährt? Der Hintergrund dafür ist relativ einfach. In Tim-Berners-Lee ursprünglichem Entwurf der HTML Version ging es hauptsächlich um den Austausch von Dokumenten (meist wissenschaftlich). Das heißt der Fokus lag bis dato nur auf die Veröffentlichung von Texten.

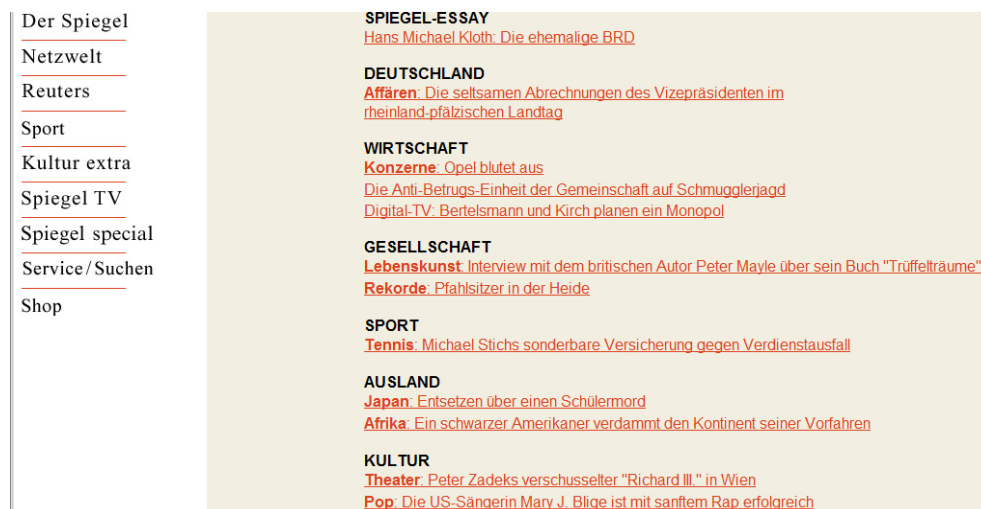


Abb. 3 - Screenshot von der Spiegel Online Webseite aus dem Jahre 1998

Erst im Laufe der Jahre kamen dann Elemente zur Implementierung von Tabellen, Bilder, Frames und Stylesheets hinzu. Seit der Veröffentlichung von Version 4.01. bis heute, kamen kaum noch zusätzliche Attribute hinzu, die den HTML Code erweiterten. Das soll heißen, dass wir heute noch mit einem Webstandard arbeiten, der eigentlich nicht unserer Zeit angepasst ist. Das Internet ist viel komplizierter geworden als das HTML unseren Ansprüchen alleine genügt. So machte es erst Flash oder Javascript oder viele andere Programmiersprachen und Anwendungen möglich, Webseiten dynamischer und multimedialer darzustellen.

HTML5 sollte jedoch eigentlich gar nicht entwickelt werden. Denn das W3C, welches die Standards für das Web festlegt, hatte sich auf die Weiterentwicklung von XML konzentriert, und

Hauptteil

nie geplant eine Neuauflage von HTML zu veröffentlichen. Erst große Firmen wie Mozilla Foundation, Opera Software ASA und Apple Inc. rebellierten dagegen und schufen ihre eigene Arbeitsgruppe - die *Web Hypertext Application Technology Working Group*⁷ (WHATWG).



Abb. 4- Screenshot von der heutigen spiegel.de Webseite

Ziel ist es die Entwicklung und Erweiterung von neuen Web-technologien. Nachdem das W3C einsah, dass keiner der nennenswerten Browserhersteller an einer neuen XML Version interessiert war und der Internet Explorer damit eh nicht wirklich umgehen kann, beschlossen sie ihre Arbeiten an den Projekt einzustellen und eine eigene HTML5 Arbeitsgruppe zu schaffen. Der große Unterschied an den zwei Gruppen ist die Veröffentlichung bzw. offizielle Nutzung von HTML5. Die WHATWG plant den Recommendation-Status nicht vor 2022, während die W3C das schon ab 2010⁸ tut. Der Recommendation-Status wird erst vergeben, wenn zweifelsfrei klar ist, das HTML5 überall problemlos eingesetzt werden kann. Das erfordert jedoch laut

⁷ <http://www.whatwg.org/>

⁸ Peter Kröner, 2010, Seite. 15

Hauptteil

Schätzung der WHATWG mindestens 20.000 Einzeltests die in mehreren Browsers funktionieren müssen. Das Jahr 2022 erscheint von daher nicht als übertrieben oder unrealistisch. Natürlich befindet sich die neue Version immer noch in der Entwicklungsphase und so wird das fast 800 Seiten lange PDF mit allen wichtigen Informationen und Spezifikationen von HTML5 ständig aktualisiert. Jedoch kann man heute schon einige dieser Elemente ohne Bedenken einsetzen, sofern moderne Browser sie unterstützen.

Doch was ist nun eigentlich so besonders und neu in HTML5?

In der neuen Spezifikation findet man eine Menge von neuen Elementen, einigen alten mit überarbeiteter Funktion und manche tauchen gänzlich nicht mehr auf. Ein Teil beschäftigt sich mit dem komplexen Aufbau von Webseiten, die vom Code her wesentlich übersichtlicher strukturiert werden sollen.

Hilfreiche Funktionen wie z.B.:

- Plugin-unabhängiges Einbetten von Multimedia-Dateien mit `<video>` und `<audio>`
- Webtechnologien wie die Barrierefreiheitsrichtlinie WAI-ARIA, MathML und SVG-Vektorgrafiken wurden adaptiert und zum Standard erklärt
- Hinzugefügt wurden neue JavaScript APIs wie z.B. einer Drag&Drop Funktion und eine Pixel für Pixel programmierbare Malfläche

Das ist zwar nur ein kleiner Auszug von dem was HTML5 zu bieten hat, jedoch sind die oben genannten Erneuerungen nichts weltbewegendes. Sieht man aber die komplette Menge der Features und die immer leistungsstärkeren Browser an, bietet sich eine Vielzahl von Möglichkeiten die einen wesentlichen Unterschied zu HTML 4.01. darstellen.

Webentwickler soll vor allem die Arbeit erleichtert werden, indem man eindeutige Elemente zur besseren Strukturierung einsetzt oder auch Teile, z.B. Tags schließen `</...>` als überflüssig

abgestempelt.

Ein simples HTML5 Dokument könnte also so aussehen:

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <title>HTML5</title>
</head>
<h1>Das ist ein HTML5 Dokument</h1>
<p class="beispiel"> Hallo!
<p> Das hier ist nur ein Beispieltext
```

Das selbe Dokument in HTML4.01.:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>HTML4.01.</title>
    <meta http-equiv="Content-type" content="text/
html; charset=utf-8" />
  </head>
  <body>
    <h1>Das ist ein HTML4 Dokument</h1>
    <p class="beispiel"> Hallo! </p>
    <p> Das hier ist nur ein Beispieltext </p>
  </body>
</html>
```

Erfahrene Webentwickler sollten sofort hier den Finger heben und meckern. Doch speichert man diese paar Zeilen Code ab und lädt sie in einem HTML5-fähigen Browser, würde man das gewünschte Ergebnis ohne Fehler erhalten. Was zunächst Auffällt ist das einige Tags wesentlich kürzer sind und manche gar nichtmehr erscheinen. So ist es nun komplett überflüssig das ein Dokument `<html>`, `<body>` und sogar `<head>` enthalten muss. Tags zu schliessen, für Tabellen oder Text Elemente,

Hauptteil

werden wir in HTML5 nur selten zu sehen kriegen, da der Parser das automatisch erledigt und es sonst nur noch überflüssige Schreibarbeit wäre. Ebenfalls überflüssig, sind auch lange Schreibweisen nur damit sie formal korrekt erscheinen.

```
<input type="checkbox" checked="checked" />  
<input type="checkbox" checked="checked">  
<input type="checkbox" checked>  
<input type=checkbox checked>  
<INPUT TYPE=CHECKBOX checked>  
<INPUT type=checkbox checked=checked />  
<InPut tYPe=cHeCKboX CheCkeD>
```

All diese Schreibvarianten sind in HTML5 zulässig. Entscheidet man sich nun für die Kürzeste könnte man damit nicht nur Zeit sparen, sondern auch Kilobyte in der HTML Datei. Weglassen von Tags und kurze Schreibweisen steigern also die Performance und verkürzt die Ladezeit, in der eine Seite angezeigt wird. Allerdings wird sich das erst bei großen und komplexen Seiten als benutzerfreundlich auswirken.

Offline-Anwendungen

Was machen nun Webapplikationen wenn die Internetverbindung unterbrochen ist? Sie funktionieren ganz einfach nicht mehr. Das ist der wunde Punkt, der Online-Webapplikationen bisher unattraktiv dastehen lässt. Habe ich meine ToDo-Liste bei iGoogle oder meinen Kalender nicht immer verfügbar, so gibt es keinen Anlass die Anwendungen zu nutzen. Ergo, ich speicher meine Daten immer lokal ab oder greife zum Zettel und Stift. Mit HTML5 und anderen Technologien hat man die Möglichkeit diese Funktionen ohne Einschränkungen vollständig offline zu nutzen. Demnach wird beim ersten Aufruf einer Internetseite, der Browser gezielt Ressourcen vorladen und cachen (z.b. HTML Seiten, Bilder oder Skripte). Sollte der Benutzer alle benötigten Daten runter geladen haben, kann er ohne weiteres offline gehen und die Webapplikation weiterhin benutzen. Sollte er in der Zeit neue Datensätze anlegen oder verändern, so werden die Aktualisierungen im Browser automatisch zwischen

Hauptteil

gespeichert und beim späteren Onlinebetrieb synchronisiert. Dem Benutzer kann es von daher völlig egal sein, ob er gerade mit dem Internet verbunden ist oder nicht.

Umsetzbar wird dieses Prinzip mit einem *Manifest*. Eine Cache Manifest Datei, ist eine normale Textdatei, in der alle benötigten Skripte usw. geladen werden, die der Browser für die einzelnen Webseiten bereithalten muss. Im Grunde genommen funktioniert das nicht anders, als die gewöhnliche Cache-Funktion des Browsers, nur werden hier die Inhalte für die Webseiten bzw. Webapplikationen freigegeben, versioniert und automatisiert, damit es zu einem reibungslosen Datenaustausch kommt (Synchronisation). Diese zwei verschiedenen Cache Methoden sind von einander komplett unabhängig und haben keinen Einfluss auf den Dateninhalt der anderen Methode.

Natürlich macht es aber auch keinen Sinn jede mögliche Webapplikation für den Offline-Modus zu cachen. Nimmt man ein Forum oder Blog als Beispiel, so müsste man alle Einträge aller Benutzer im Manifest cachen damit es einwandfrei angezeigt wird. Die dafür benötigte Datenmenge wäre dann um ein vielfaches höher als gewollt und ein weiterer Nachteil. Hätte der Benutzer nun in der Phase ohne Internetverbindung auf Fragen oder Themen geantwortet, wären eventuell Nachrichten nach der Synchronisation nicht mehr in der zeitlich oder sinngemäßen korrekten Abfolge. Themen oder Beiträge könnten in der Zeit schon nicht mehr existieren.

Um solche Fälle zu vermeiden ist es ohne weiteres möglich mit Hilfe der *Online-Whitelisting* Funktion bestimmte Ressourcen zu sperren und nur in Verbindung mit dem Netzwerk zugänglich zu machen. Wer nun befürchtet das einige Webseiten mit den genutzten Webapplikationen deutlich langsamer geladen werden, den kann man entwarnen. Die Funktion *swapCache*, die die alten oder noch nicht vorhandenen Datensätze aktualisieren soll tritt erst in Kraft, nachdem die Seite komplett angezeigt wird. Nach erfolgreicher Synchronisation wird der Benutzer lediglich gebeten die Seite eventuell neu zu laden, da es sich noch um eine alte Ansicht handeln kann.

Multimedia Elemente und eingebettete Inhalte

Betrachtet man heute moderne Webseiten, so stellt man schnell fest, dass die meisten multimedial sind – Videos, Animationen, Sounds usw. Aber um diese Elemente auch auf einer Seite einzubauen kommt man um Adobe's Flash Technologie nur schwer bis gar nicht vorbei.

Denn HTML hatte damals nie vorgesehen, solche komplexen Medien übers Internet zu verbreiten. So ist es bis heute nicht möglich, ohne Flash Plugin, Medienelemente im herkömmlichen HTML Code zu integrieren. Die Entwickler von HTML5 haben sich darüber einige Gedanken gemacht und stellen die Tags `<audio>` und `<video>` zur Verfügung, die das Einbetten von Multimedia-Elementen ohne extra Plugin bewerkstelligen sollen.

Mit `<audio>` lassen sich Audiodateien oder Audiostream in die Webseite implementieren. Mit dem Attribut *controls* fügt man Steuerelemente (Play, Pause, Stop etc.) hinzufügen um eine Bedienoberfläche zu erhalten. Ist einem jedoch das eigene Interface lieber, wäre es weiterhin möglich das mit Javascript umzusetzen.

Mit dem `<video>` Tag lässt sich das Ganze ähnlich behandeln. Es verfügt ebenfalls über Steuerelemente und via CSS kann die Höhe und Breite des Videos bestimmt werden, sofern keine anderen Vorgaben vorliegen. Nützliche Funktionen wie `<preload>` sorgen dafür, dass Videos sofort beim fertigem Laden abgespielt werden. `<posters>` hingegen kann als Platzhalter für ein Image dienen, das angezeigt wird, sollte die Quelle des Videos nicht vorhanden oder fehlerhaft sein.

Mit dem *source* Attribut, welche beide Medienelemente benötigen, lassen sich nun mehr Möglichkeiten anstellen. So kann man den Tag ausklammern und individuell für jede Art von Benutzer definieren.

```
<video>
  <!-- Für Handybenutzer - ->
  <source src="video_klein.avi" media="mobile">
  <!-- Alle Anderen - ->
  <source src="video_normal.avi" media="all">
</video>
```

Desweiteren könnte man die Quellen anhand des benutzen Codes sortieren, da diese vom Browser abhängig sind und stark variieren.

So benutzt z. B. Safari, Chrome und Internet Explorer 9 den Codec H.264, der jedoch nicht in Firefox, Opera oder älteren IE's unterstützt. Die verschiedenen Softwarepatente und Lizenzen der Browserhersteller die für den Codec-Wirrwarr sorgen, lässt es also nicht zu sich auf einen gemeinsamen Nenner zu kommen.

Die „Allround“ Variante wäre dann:

```
<video controls>
  <source src="video1.mp4" type="video/mp4">
  <source src="video2.ogg" type="video/ogg">
</video>
```

Über den Browser könnte man den vorhandenen Codec auslesen und nur die gewünschte Videodatei laden oder streamen. Nachteil: Videos müssten mehrmals unterschiedlich codiert werden.

Drag & Drop API

Die Drag & Drop Funktion wurde ursprünglich von Microsoft entwickelt und hatte den ersten Auftritt damals im Internet Explorer 5, der jedoch mächtig in die Hose ging. Denn das Hauptproblem der API liegt an diversen Funktionen und Events, die nicht das tun, was man bisher aus der täglichen Javascript & DOM Praxis kennt. Auf der anderen Seite jedoch sind mit ihr Dinge möglich, die mit Javascript und Co. kaum umsetzbar sind. Der Grund für die Aufnahme und Standardisierung erscheint dann zunächst fraglich, aber Safari und Mozilla hatten diese Funktion schon in ihren Browsern integriert und drängten nun

darauf hin die API in HTML5 aufzunehmen.

Ursprünglich sollte man markierte Texte und Bilder umherziehen können oder das Schieben von Links in die Adresszeile ermöglichen. Wären die Ansprüche an die API heute immernoch so gering, gäbe es sicherlich kaum Verwendung dafür, auch wenn eine Drag & Drop Funktion, wie man sie aus Windows kennt, sehr benutzerfreundlich ist. Stattdessen erfand Ian Hickson, Google Mitarbeiter und Vorsitzender der WHATWG, das HTML Attribut *draggable*, welches es ermöglicht jedes beliebige Element mit dem Zusatz *draggable* - „ziehbar“ zu machen. Wird nun die Drag&Drop API als Standard spezifiziert und in allen Browsern implementiert, so wären alle Browser untereinander interoperabel.

Folgende Anwendungsbeispiele:

- Browser A biete die Möglichkeit das Verschieben von JPEG Dateien zu zulassen, während Browser B für dessen Empfang gerüstet ist - folglich sind diese beiden Browser kompatibel zu einander.
- Desktop- und Onlineanwendung könnten damit ebenfalls verknüpfen werden. Nicht nur simple Bilddateien, sondern komplette Datensätze werden ziehbar und zwar kreuz und quer über alle Browser inklusive Desktop.
- Angenommen, ein Benutzer erstellt sich im Forum ein Profil und braucht noch ein passendes Bild für einen Avatar, müsste man mit einer Upload Funktion nach lokalen Bildern suchen. Mit Drag & Drop lässt sich so ein Bild von Facebook, studiVZ etc. ziehen und problemlos einfügen.

Bisher hat nur Firefox ab Version 3.5 aufwärts die Drag & Drop Funktion übernommen. Ziehen die anderen Browser gleich, so steht allen Anwendungsbereichen nichts mehr im Wege.

Das Canvas Element

...zu Deutsch “Leinwand”, gehört zu den spektakulärsten Neuerungen und ist wohl das beeindruckendste Element, wenn man von HTML5 spricht. Im Grunde genommen stellt uns `<canvas>` eine Fläche zur Verfügung, die Mittels JavaScript, Pixel für Pixel genaue Bitmap-Grafiken erstellt. Ein programmierbares `` also, mit dem sich Diagramme, Animationen, Interface-Elemente kreieren und Videos manipulieren lassen.

2D Bereich

Neben den Zeichnen von eigenen Bildern, lassen sich auch beliebige Bildformate auf eine `<canvas>`-Ebene importieren, skalieren, negativ darstellen etc. Das eigenst programmierte Kunstwerk lässt sich anschliessend auch problemlos als pixelbasierte Grafik exportieren. Spinnt man die ganze Idee weiter, so wären Webapplikationen in der Art eines Bildbearbeitungsprogramm, im Browser keine weit hergeholte Idee. Das Hinzufügen von Steuerelementen und Funktionen via JavaScript lässt sich mit genug Programmierkenntnissen einbauen. Natürlich wäre es nicht mit Adobe’s Photoshop und dessen kompletten Umfang vergleichbar, aber mit Paint sollte es alle Male mithalten können.

Da sich das `<canvas>` Element bisher schon großer Beliebtheit bei den Webentwicklern im Umgang mit HTML5 erfreut, gibt es mittlerweile genauso viele Frameworks und vorgefertigte JavaScript Plugins auf dem Markt, wie Sand am Meer.

Ein Beispiel von Sinnvollen Projekten:

- Das Schriftersetzungsprojekt Cufón⁹ nutzt `<canvas>`, um Texte in Schriften darzustellen, die nicht wie übliche Schriften auf allen Betriebssystemen vorzufinden sind
- Processing, ist eine JavaScript-Portierung die gezielt bei Grafik und Animationen eingesetzt wird. Benutzt ebenfalls `<canvas>`

⁹ <http://github.com/sorccu/cufon/wiki/about>

Hauptteil

- Mozilla Foundation hat mit dem Projekt: Bepin - eine komplette Programmoberfläche mit <canvas> umgesetzt

Bepin¹⁰ ist ein Code-Editor der in JavaScript geschrieben wurde und sich individuell anpassen lässt. Da es eine Online Webapplikation ist, können z.B. mehrere Entwickler an unterschiedlichen Orten gemeinsam an einem Code arbeiten, der bei Veränderungen für alle Anwesenden in Echtzeit aktualisiert wird.



Download Tracker

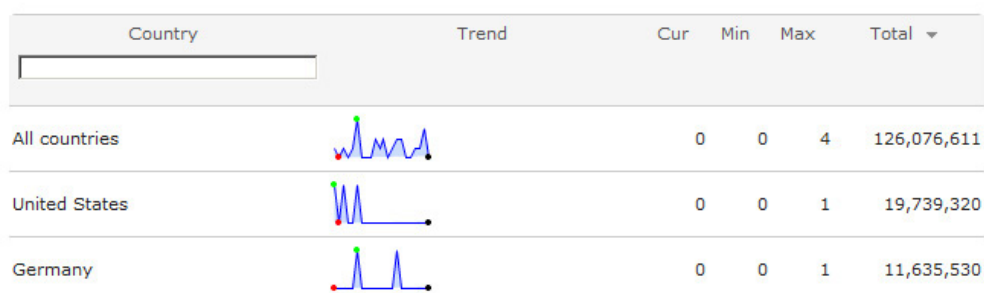


Abb. 5 - Screenshot von: <http://www.mozilla.com/en-US/firefox/stats/>

Auf der Website von Mozilla ist ein wunderbares Beispiel für die Kombination von der <canvas>-Ebene und einer Scalable Vector Graphic (SVG). Anhand der Downloads weltweit wird eine Statistik angezeigt, wo und wie oft Firefox 3.6 gerade herunter-

¹⁰ <https://bepin.mozillalabs.com>

Hauptteil

geladen wird. Die SVG Grafik, welche sich wunderbar für dynamische und interaktive Zwecke nutzen lässt, ist in dem canvas Element eingebettet, welches wiederum in Echtzeit durch die Daten vom Server aktualisiert wird. Kleine gelbe, aufblickende Punkte zeigen dann die Standorte an.

3D Bereich

Um das <canvas> Element komplett auszureizen reichen die Möglichkeiten in der Darstellung oder Animation von 2D Grafiken noch lange nicht. Die Königsdisziplin liegt in der Kreation von 3D Objekten. Wer sich jedoch auf das Gebiet wagt, muss unbedingt auf hardwarebeschleunigte APIs zurückgreifen wie z.B. WebGL. Denn die HTML5 Spezifikationen reichen bisher nur für die 2D Zeichenfunktionen. Es gibt zwar bisher ein paar Frameworks die 3D Objekte anhand von einem 2D Context umwandeln (Gittermodelle, texturierten / animierten Objekte), jedoch ist die Qualität- und Performancegrenze schnell erreicht.

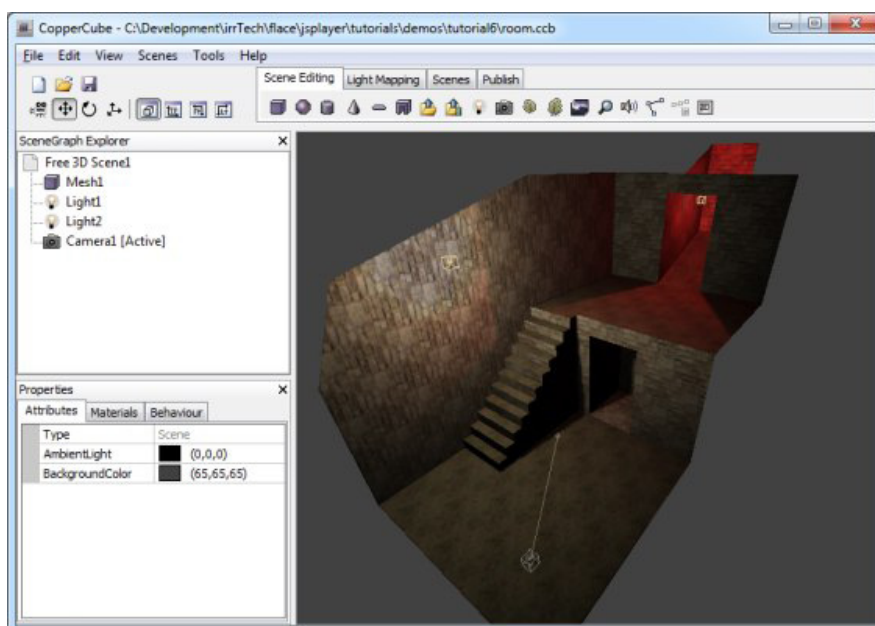


Abb. 6 - Screenshot des Leveleditors von CopperLight

Die Frameworks für das erstellen bzw. umsetzen der 3D Modelle erleichtern sollen, liefern nicht nur eine benutzerfreundliche Arbeitsoberfläche mit vielen Vereinfachungen, sondern wie

¹¹ <http://www.ambiera.com/copperlicht>

Hauptteil

im folgenden Beispiel von *CopperLicht*¹¹, sogar einen eigenen Leveleditor speziell fürs Gamedesign ausgelegt. Man kann problemlos ohne Programmierkenntnisse seine eigene Welt schaffen, mit unzähligen vorgefertigten Objekten aus der Datenbank. Wer seiner alten Umgebung lieber treu bleiben will, kann auch ohne weiteres die gängigen 3D Formate (*.3ds, *.mesh, *.obj) aus 3D Studio Max oder Autodesk Maya importieren. Das erstaunliche an diesem Beispiel ist nicht nur, dass 3D Objekte im Browser dank JavaScript erzeugt und anschliessend gerendert werden, sondern dass man mit ihnen auch interaktiv agieren kann. Im Asteroid Game kann ich mit den Tasten W,A,S,D durch den simulierten Raum bewegen und durch drücken der Leertaste die Asteroiden zerstören. Eine Kollisionsabfrage mit dem erzeugten Asteroiden und dem Spieler gibt es leider nicht, aber sobald auf die gezielten Steinbrocken geschossen wird, werden die in einer Animation zerstört.

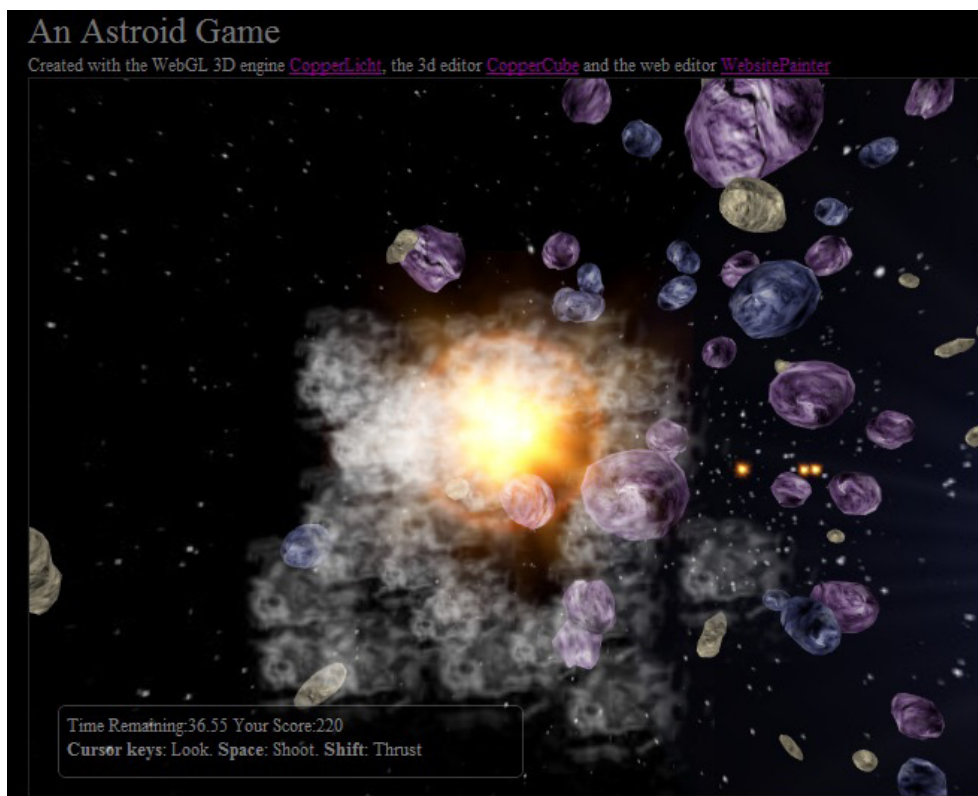


Abb. 7 - Screenshot des CopperLicht Projektex: Asteroid Game

Hauptteil

Auch wenn das hier nicht das aktuelle Maß der technischen und grafischen Möglichkeiten heutzutage ist, muss man immer im Hinterkopf behalten, dass dies alles auf HTML5 und JavaScript APIs basiert. Und somit in allen HTML5-fähigen Browsern benutzt werden kann.

Fassen wir soweit zusammen

HTML5 liefert also den Grundstein zum Einbetten solcher Medien ohne weitere Plugins oder Programme wie z.B. Flash. Damit bietet man allen Benutzern der gängigsten Browser, plattform-unabhängige Kompatibilität, sofern man einen aktuellen Browser besitzt, der die gewünschten Funktionen heute schon unterstützt. Durch JavaScript (bzw. Frameworks) werden die ganzen Prozesse geschrieben und umgesetzt. Vorteil der ganzen Geschichte ist: JavaScript ist ein ständiger Bestandteil jedes ernstzunehmenden Webbrowsers, und bei fast allen Betriebssystemen in der Grundinstallation schon dabei. Ob nun Firefox, Safari, Windows, Mac oder Linux - JavaScript ist überall zu finden. Man kann ohne weiteres behaupten, dass JavaScript sich zu denen an wichtigsten Plattformen entwickelt. Denn bisher war GoogleMail, das mit komplexester was JavaScript alleine zu bieten hat. Erst in Verbindung mit z.B. WebGL werden dann die nötige Schnittstelle geliefert, um auch aufwendige Anwendungen umzusetzen.

Wie sieht nun die Zukunft von HTML5 aus ?

Wenn man die neuen Spezifikationen und Features von HTML5 betrachtet, fällt ein deutlicher Unterschied zu HTML 4.01. auf. Lag der Fokus damals auf das Verbreiten von Bild und Text, so änderte er sich nun in die Richtung von Webapplikationen. Das Ganze kommt nicht von irgendwoher, denn der ursprüngliche Name von HTML5 lautete "Web Applications 1.0". Der Verbindung von HTML4.01., CSS und JavaScript fehlt es an vielen Features, die für Webapplikationen benötigt werden - HTML5 schließt diese Lücken.

Die eigentliche Idee von HTML5 war den Browser als einfacher

Hauptteil

Dokumentenbetrachter in eine universelle Anwendungsplattform zu gestalten, und zwar für alle Arten von Programme. Der Browser dient heute nicht mehr nur zum Ansehen von Webseiten, sondern wird zum Teil schon als Mediaplayer, Laufzeitumgebung von 3D Shooter, Grafikbearbeitungsapps und Officeanwendungen genutzt. Jedoch alles bisher mit Plugins. Auf den Höhepunkt der Vision hat es Google mit seinen zukünftigen Betriebssystem - Chrome OS geschafft. Das der Trend immer mehr zu Webapplikationen tendiert, kommt auch nicht von irgendwoher, denn die WHATWG wird gerade von den Firmen gesteuert, deren Tätigkeit sich auf Webanwendungen beziehen. Google hat natürlich ein großes Interesse daran, das der Browser von Morgen wesentlich mehr kann, welches dem Prinzip ihres Betriebssystem ähnelt.

Nun ist der vorgeschlagene Recommendation Status von der WHATWG aber erst in 12 Jahren, in der Zeit kann im Bereich des Webs noch so einiges passieren, voralldem da schon HTML6¹² angekündigt ist. Obwohl schon heute HTML5 genutzt wird und das Internet ein Stück weit revolutioniert, kann man bisher keine genauen Aussagen treffen ob HTML5 wirklich zum neuen Standard deklariert wird.

2.3.2 WebGL

Eine Arbeitsgruppe der Khronos Group¹³, ist ein zusammengefügter Haufen voller Programmierer von Medienunternehmen, die Standards und APIs entwickeln. Unteranderen haben sie auch damals das OpenGL entworfen und auf den Markt gebracht.

WebGL gehört heute zu den wichtigsten 3D-Grafik-Programmierschnittstelle für Webbrowser, das auf der Basis von OpenGL ES im Zusammenspiel mit JavaScript, aufwendige 3D-Anwendungen im Browser rendert. Google hatte einst seine eigene Schnittstelle namens O3D entwickelt, welche sehr vielversprechend aussah. Die Arbeit wurde jedoch letztlich eingestellt um sich an der Weiterentwicklung von WebGL zu beteiligen¹⁴.

¹² <http://webstandard.kulando.de/post/2010/01/14/html6-the-next-generation>

¹³ <http://www.khronos.org/>

¹⁴ <http://www.browser-fuchs.de/?p=1087>

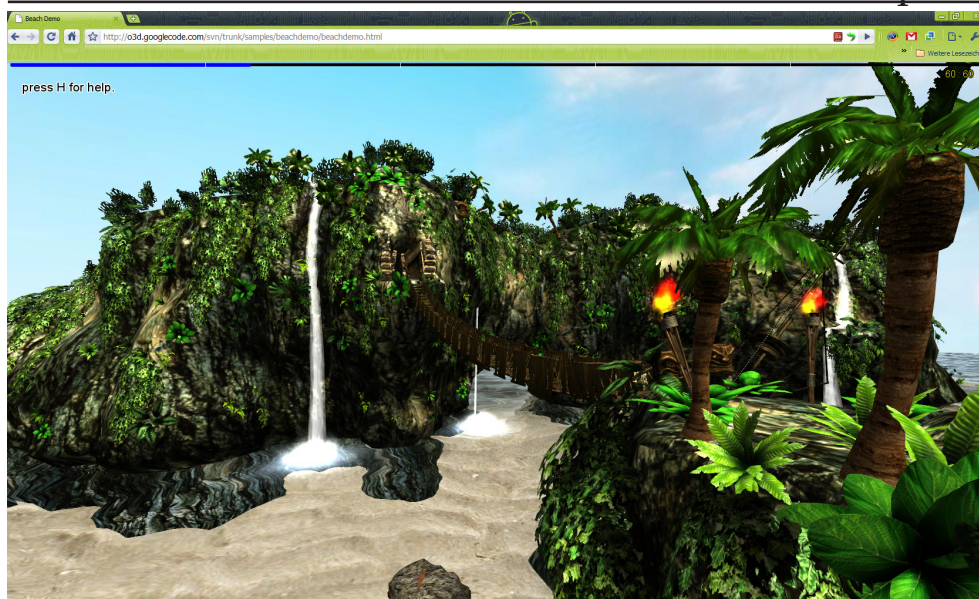


Abb. 8 - Livedemo von Google's O3D Schnittstelle - Beache

2.3.3 CSS Level 3

Wer von HTML spricht, hat auch meistens gleich die Cascading Style Sheets (CSS) im Hinterkopf. Denn ohne die Ergänzung des HTML Codes mit CSS, würde das Internet ziemlich trist und fade aussehen.

1993 kamen damals die ersten Vorschläge zur Erweiterung des Webs mit CSS. Erst nach der Zusammenarbeit von Håkon Wium Lie und Bert Bos wurde konkret an einer tauglichen CSS Version gearbeitet, so wie wir sie heutzutage kennen. Im folgenden Jahr wurde auch das World Wide Web Consortium drauf aufmerksam und veröffentlichte im Dezember 1996 die CSS Level 1 Recommendation. Die folgenden Versionen wurden von vielen Browsern unterschiedlich interpretiert, so das bald CSS Level 2.1 publik wurde, welche Webentwickler derzeit nutzen. Seit dem Jahr 2000 ist die neue Version CSS3 in der Entwicklung und wird im Vergleich zu den älteren CSS Versionen modular aufgebaut sein.

Einiges was für HTML5 zutrifft, gilt auch für CSS3. Viele der Features sind schon umgesetzt, die das Strukturieren und vor

Hauptteil

allem das Designen wesentlich vereinfachen. Mit neuen Elementen und Attributen, wird für Mozilla und die Webkit Engine, auf Kompabilität der unterschiedlichen Browser gesetzt. Simple Animationen, für beispielsweise eine Menünavigation, war in CSS2.1 kaum bis gar nicht machbar, und nur umsetzbar durch Flash, jQuery oder JavaScript. Doch das hat sich nun geändert.

Das wohl am ehesten alltagstaugliche Feature wird wohl „Multi-Columns“ sein. Internetseiten wie z.B. Wikipedia, Finanzial Times oder andere etliche Zeitungen und Magazine werden dadurch wesentlich einfacher gestaltet sein, was für den Benutzer vor dem Monitor eventuell gar nicht sofort sichtbar ist.

Mit *Multi-Columns* lassen sich Texte anlegen und in mehreren Spalten je nach Breite aufteilen (die Barrierefreiheit ist immernoch gewährleistet). Bisher haben die Content Managment Systeme übernommen, jedoch nicht mit wirklich umfangreichen Funktionen zur Gestaltung. Sehe ich mir z.B. Seiten von Zeitungen mit einer hohen Auflösung an so legt mir CSS die Texte wie gewünscht drei spaltig an. Bei einer niedrigeren Auflösung oder beim Zusammenschieben des Browsers, wird es auf zwei- bis einspaltig formatiert.

Slideanimation, Verläufe, runde Ecken und Schattierungen lassen sich relativ schnell und simpel in CSS3 umsetzen ohne das Benutzen von extra Grafiken. Bei Webseiten mit viel eingebauten Grafiken, sollte sich die Performance bzw. das Laden solcher Seiten deutlich erhöhen.

Durch das klicken des ersten Menüpunktes (obere Bildhälfte Abb. 9), wird in einer Animation wie wir es bisher nur von Flash oder jQuery kennen, der Text von Menüpunkt 1 „ausgefahren“ und der Rest nach unten verschoben.

Inspiziert wurden die Entwickler von CSS3 von genau solchen Frameworks wie jQuery die auf JavaScript basieren. Vorteil hier wieder: es werden keine Plugins nötig sein.

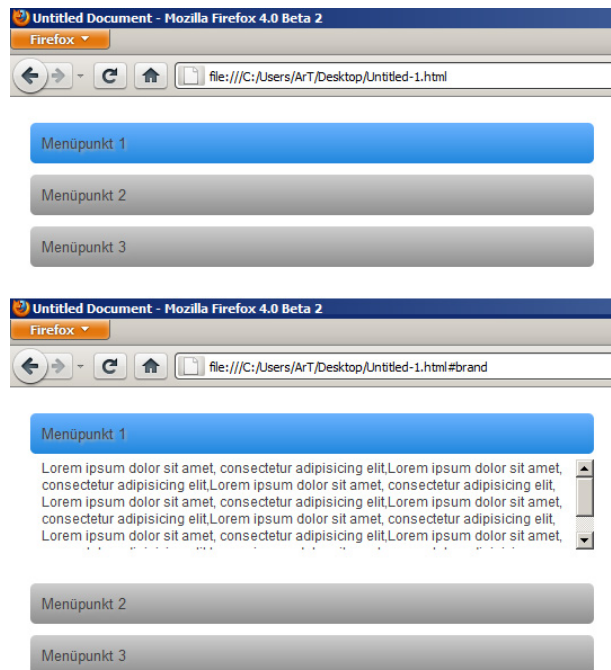


Abb. 9 - CSS3 Slideanimation im Menübeispiel

In einem Interview¹⁵ mit Eric Meyer traf er folgende Aussage-

[...]

How has JavaScript influenced CSS3 (and vice versa) and what roles do you see each one taking in modern web development; do you see them being more distinct or more blended together than they are now?

[Eric Meyer]: From what I've seen, most of the influence has been on JS frameworks like jQuery, with that influence coming from CSS. However, there is some movement going back the other way- ideas like 'contains' that were first done in JS are starting to be considered by the CSS WG.

I think the much bigger influence on CSS by JS will be in people using JS to „silently“ add advanced CSS to browsers. I wrote about this recently, and I'm pretty sure it's just a matter of time before it actually happens. We're already seeing things like using JS to natively turn browsers into speaking browsers, and to hack around ancient accessibility barriers. Adding support for HTML5 elements and CSS3 features via JS is not far behind.

[...]

¹⁵ <http://sixrevisions.com/interviews/six-questions-eric-meyer-on-css3/>

CSS Level 3 erweitert seinen Bereich in Richtung Z-Achse

Matrix3d, translate3d, scale3d, rotate3d, perspective und back-face-visibility sind bisher für Webentwickler völlig unbekannte Attribute, die in ihrem bisherigen Arbeiten mit CSS 2.1 nicht auftauchten. Der Grund dafür ist ziemlich einfach, denn erst mit CSS3, wird der Arbeitsbereich um eine Achse erweitert. So lassen sich nun durch reines HTML und CSS 3D-Objekte im Browser erzeugen und darstellen.

Wer dann noch das Ganze mit flüssigen Animationen kombinieren will, kommt um ein bisschen JavaScript Code nicht herum.

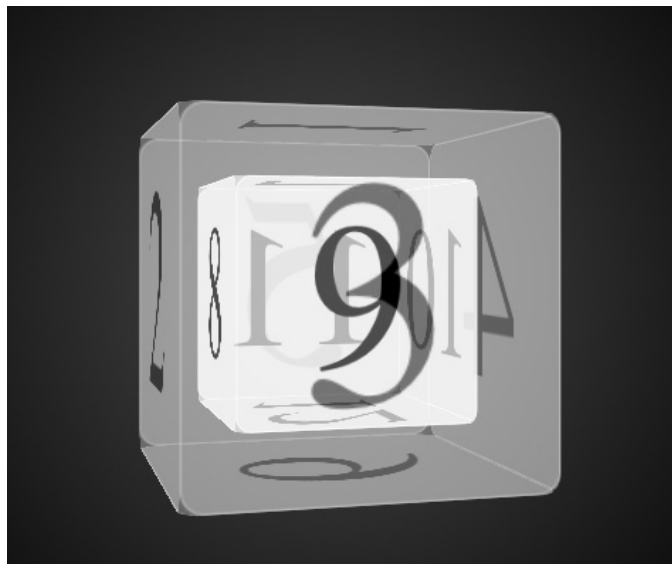


Abb. 10 - 3D-Würfel-in-Würfel mit CSS generiert

Die Darstellung solcher Objekte und Animationen bleibt jedoch bisher nur den Webkit Browsern vorenthalten und ist derzeit noch nicht für Mozilla Firefox oder Google Chrome verfügbar. Die Entwickler nutzen dabei den Entwurf des *CSS 3D Transform Module Level 3*, der damals im März 2009 von Apple-Entwickler als WC3-Working-Draft eingereicht wurde. Auf dem iPhone lassen sich solche Spielereien schon seit der Betriebssoftware Version 2.0 realisieren.

Bisher wurde keiner dieser Entwürfe vom W3C als Standard publiziert und ist deswegen bisher nur Benutzer der aktuellen Safari-Browser zugänglich.

Hauptteil

Das bisher am gelungenste interaktive 3D Projekt, was nur anhand von CSS3 und HTML umgesetzt wurde, lautet *Snow Stack*¹⁶.

Das Projekt ist zwar keine innovative Neuerfindung, da es eine Nachmachung des bereits vorhandenen Plugins Cooliris¹⁷ ist, kann es jedoch ohne extra Installation auskommen.

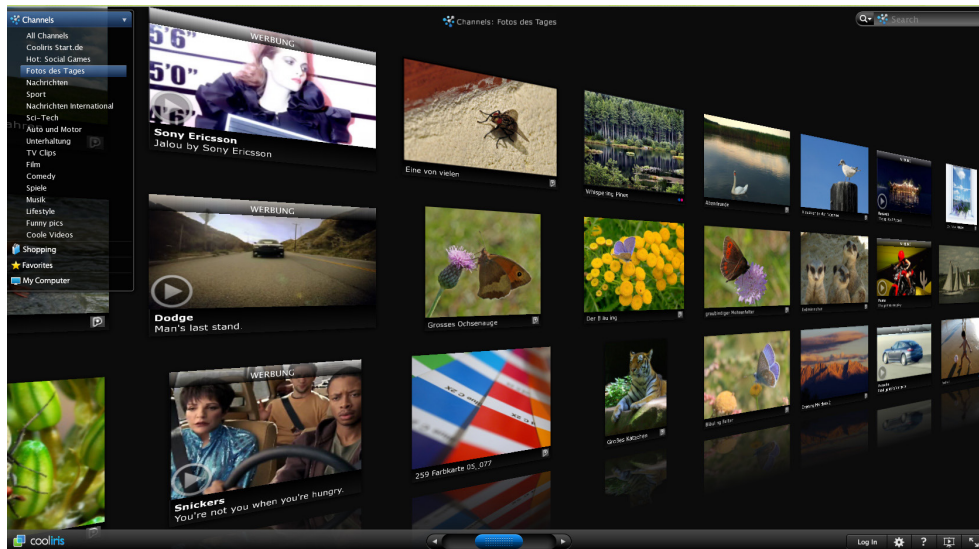


Abb. 11 - Screenshot von Cooliris

Einfach gesagt, ist Cooliris eine räumlich-interaktive Gallery zum betrachten von Foto- und Videomaterial. Das Plugin kann man sich seit einigen Jahren kostenlos runterladen und installieren. Es erweitert den Browser um die Fähigkeit, sich zum Beispiel Bild-Suchergebnisse bei Google in einer, theoretisch unendlich langen Leinwand, anzugucken. Beliebige Bilder können angeklickt werden zum vergrößern oder speichern. Auch Videos die eventuell auf YouTube & Co. verlinkt sind, können in Cooliris abgespielt werden. Das Ganze wurde mit Hilfe von Flash AS3 und PV3D programmiert. Das setzt natürlich auch voraus, dass jeder Benutzer das Adobe Flash- und Cooliris-Plugin installiert haben muss. Snow Stack hingegen wird nicht Software von Drittanbietern benötigen um im Browser einwandfrei zu funktionieren, da das komplette Projekt mit CSS umgesetzt wurde.

Während meiner Recherche über das Thema CSS 3, speziell mit den neuen 3D Features, kamen häufig Anmerkungen und Fra-

¹⁶ <http://www.satine.org/archives/2009/07/11/snow-stack-is-here/>

¹⁷ <http://www.cooliris.com/tab/#c=322&f=Channels>

Hauptteil

gen, ob man nun dank CSS3 Flash komplett ablösen kann. Das ist natürlich keine Frage die man jetzt sofort beantworten kann. Aber unter Betrachtung einiger Fakten kann man eventuell einen klaren Trend erkennen, der nicht nur Abhängig von den neuen Features ist, sondern auch durch Konzerne und Firmen beeinflusst wird.

Was spricht also für die Verdrängung von Flash...

- Animationen die bisher in Flash erstellt werden mussten, aber nun auch in CSS3 umsetzbar sind, wird man wegen der Performance und Kompabilität für Nicht-Flash Nutzer umschreiben oder zukünftig nur noch in CSS3 erstellen
- Webentwickler, die nicht auf Flash spezialisiert sind, werden definitiv auf die CSS Varianten zurückgreifen, sofern möglich. Das beinhaltet auch den privaten Bereich (Webseiten) von Hobby-Designer und Entwickler
- Mit Frameworks wie jQuery & Ajax (basierend auf JavaScript) sind teilweise Animationen möglich wie mit Flash -> keine Plugins nötig
- Großkonzern Apple hat bekannt gegeben das weder auf dem iPhone noch iPad jemals Flash unterstützt wird. Das entspricht Apples Firmenpolitik - sie setzen lieber auf HTML5 für mehr Kompabilität
- Auch Microsoft konkurriert mit der hauseigenen Marke „Silverlight“ gegen Adobes Flash an

Fakten die gegen eine Abschaffung von Flash zählen...

- Flash ist ein wichtiger Bestandteil von etlichen Webseiten und wird nicht in absehbarer Zeit ersetzt oder ausser Acht gelassen

Hauptteil

- Google und Adobe haben sich mit einigen Firmen zusammen geschlossen unter den Name „open handset alliance“¹⁸ mit der sie gemeinsam die Mobile-Software Andoid verbessern, in der Flash einen festen Platz hat
- Viele Features die derzeit mit Flash möglich sind, kann man mit HTML5 und CSS3 bisher nicht umsetzen
- Um alle Funktionen von HTML5 und CSS3 nutzen zu können, müssen diese auch von allen Browsern einheitlich gelesen werden. Da beide noch in der Entwicklung stecken und keins bisher durch die W3C noch der WHATWG zum festen Standard deklariert wurde, ist eine zeitnahe Ablösung von Flash nicht zu erkennen
- Während immernoch etliche Werbeagenturen auf der Welt, Webseiten für das Sorgenkind „Internet Explorer 6“ optimieren und zurecht rücken, so könnte man anhand der Trägheit dieser Entwicklung von ausgehen, dass das gleiche mit Flash passiert -> es wird nicht von heute auf morgen verschwinden

Vergleicht man nun beide Seiten, so bleibt die Frage für mich ziemlich offen, ob CSS3 und HTML5 zusammen nun Flash komplett verdrängen wird. Weder HTML5 noch CSS3 ist bisher so weit im Web verbreitet, das man daraus schon feste Schlussfolgerungen ziehen kann, auch wenn große Portale wie z.B. YouTube schon auf HTML5 umgestiegen sind.

Die Entwicklung einer komplett neuen Technologie oder Sprache, die mindestens den gleichen Zweck wie Flash erfüllen und darüber noch hinaus, kann sich in den nächsten Jahren ebenfalls bewahrheiten. Sieht man sich aber diese überaus starke Kombination aus HTML5, CSS3 und JavaScript an, so kann man behaupten das es um Flash nicht allzugut steht.

Fazit: Flash ist noch lange nicht tot, aber es liegt definitiv im sterben.

¹⁸ <http://www.openhandsetalliance.com/>

2.3.4 Hardwareunabhängiges Medium...aber wie?

HTML5 und CSS3 und ein ganzer Sack, voller neuer und wirklichen guten Features, revolutionieren den Browser ein Stück weit. Die Performance im 2D Bereich sollte sich leicht verbessern und der 3D Bereich bietet viele neue Möglichkeiten, zum Gestalten von neuen interaktiven Anwendungen oder Webseiten. Doch das ganze macht den Browser weder hardwareunabhängiger, noch kann man ihn schon als eigenständiges Medium ansehen. Fügt man dem ganzen jedoch noch zwei Komponenten hinzu - „Streaming Media + Cloud Computing“, so sieht das Ganze schon wieder etwas anderes aus. Doch wie funktioniert das Alles ?

2.3.5 Streaming Media - die neue Generation

Was man bisher unter dem Begriff „streamen“ verstand, war das verschicken von Audio- oder Videodaten von einem Rechnernetz zum heimischen Computer, der die übermittelten Daten sofort wiedergibt. Dabei werden die Daten nicht lokal vorher gespeichert und dann erst abgespielt. Lediglich eine kleine Datenmenge wird temporär abgelegt und dient als Puffer, denn falls die Verbindung zu langsam sein sollte, würde das einer flüssigen Wiedergabe im Wege stehen. Das ganze Prinzip klingt ziemlich gleich mit dem des Rundfunks, ist es jedoch in keinsten Weise. Denn während beim Rundfunk ein Sender von vielen Empfängern empfangen werden kann, wird beim Streamen für jeden Benutzer die übermittelten Daten gesondert und auf dessen Anforderungen realisiert.

Bisher war es üblich Web-Radio oder Web-TV Angebote zu nutzen. Mit einem einfachen Plugin wie z.B. Flash, Quicktime etc, ist es dann möglich den Browser als Mediaplayer umzufunktionieren, was ihm um eine weitere wichtige Funktion erweitert. Die einzige Voraussetzung hierbei, ist eine relativ schnelle und konstante Internetverbindung. Sollte die Internetleitung nicht beide Eigenschaften haben, kann es sein das der Puffer jedesmal neu geladen werden muss oder die Verbindung zum Server

Hauptteil

immer wieder gänzlich abbricht, was dem Benutzer einer Übertragung keinen Spaß bereitet. Eine völlig neue Art des Streamens, ist die Übertragung von kompletten Spielinhalten.

Am 25. August 2009 wurde Quake Live der Öffentlichkeit zugänglich gemacht.

Quake Live¹⁹ ist ein werbefinanzierter Ego-Shooter, der auf das damalige Computerspiel Quake III Arena basiert das 1999 veröffentlicht wurde. Die Besonderheit von Quake Live ist, dass es komplett im Browser gespielt wird.

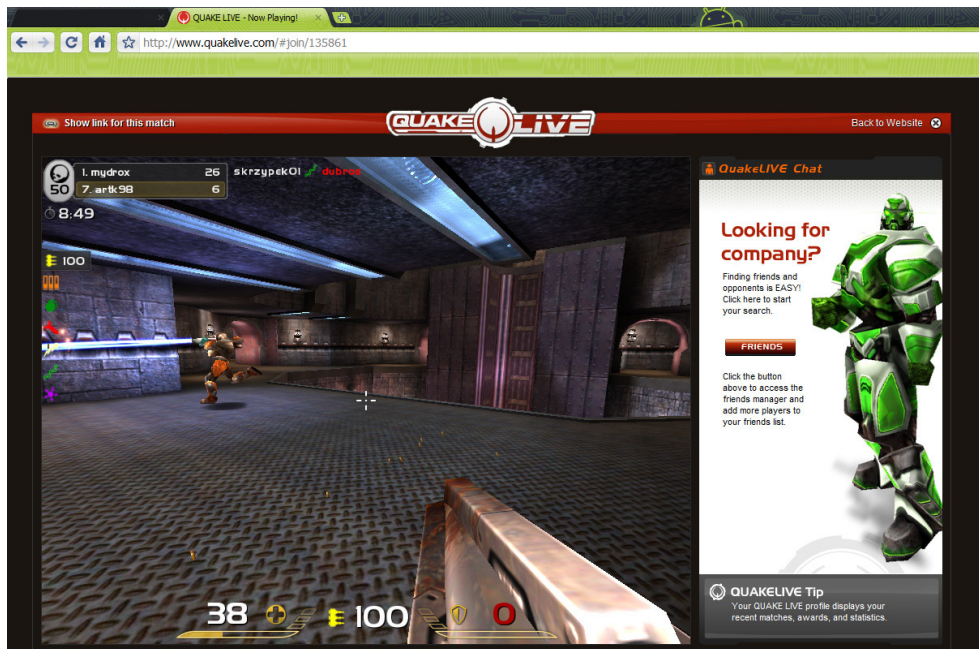


Abb. 12 - Screenshot von Quake Live im Browser

Nach der Registrierung auf der Webseite www.quakelive.com lädt man sich einen Installer runter, der das benötigte ActiveX Plugin installiert um Quake Live im Browser zu verwenden. Anschließend werden noch ca. 200mb Spieldaten runtergeladen die ebenfalls benötigt werden. Das Browser Plugin sorgt nicht nur dafür das Quake Live einwandfrei läuft, sondern das die Spieldaten auch immer aktuell gehalten werden. Bei jedem erneuten einloggen auf der Webseite, werden Updates automatisch runtergeladen. Die Hardwareanforderungen sind bei die-

¹⁹ <http://www.quakelive.com>

Hauptteil

sem Spiel sehr gering, denn trotz einer etwas überarbeiteten Grafik, ist das Spiel komplett von 1999 übernommen worden. So sollte es einen auch nicht wundern, wenn Quake Live auf dem Handy funktioniert. Der Datenaustausch vom eigenen PC zum Server von Quake Live beschränkt sich auf Positionsangaben der verschiedenen Spieler in einem Level, Kollisionsabfragen und Treffer / Lebenspunkte.

Betrachtet man Quake Live auf den ersten Blick stellt man fest das es erstmal relativ unspektakulär ist. Denn das Spiel ist nicht das aller Neuste und den Online Modus gab es ebenfalls schon einmal. Doch das innovative ist die komplette Umsetzung im Browser. Denn der dient nicht nur als eigene Plattform, sondern auch als einzige Schnittstelle des Datenverkehrs zwischen Spieler und Server, die in Echtzeit gerendert werden (60 FPS). Was bei Computerspielen oder Anwendungen bisher nötig ist, übernimmt hier der Browser - die „Softwareumgebung“. Quake Live ist also der Beweis bisher, das der Browser zu wesentlich mehr fähig ist als, nur das stupide Darstellen von Webseiten. Sofern er mit dem nötigen Extras ausgerüstet wird. Doch das ganze klingt immernoch nicht ganz nach einer hardwarefreundlichen Lösung.

Das Prinzip von Video-on-Demand, was man bisher z.B. von Telekom Produkten kennt, haben sich die Entwickler von OnLive zu Gute gemacht. Filme die man sich bisher freischalten konnte über den TV, klappt nun auch mit Games über OnLive.

OnLive²⁰ ist eine völlig neue, von der Hardware los gelösten, Plattform für Computerspiele und basiert auf dem Prinzip des Cloud Computing (dazu mehr im nächsten Abschnitt). Das bisherige Problem was die Computerhardware angeht ist einfach gesagt - sie veraltet zu schnell.

Kauf man sich heute einen neuen PC oder Laptop, so kann man ihn für aufwendige 3D Applikationen ca. zwei bis maximal drei Jahre benutzen. Will man dann weiterhin aktuelle Games spielen, muss ein neuer PC her oder man rüstet einzelne Hardwareteile auf. Um das Problem gänzlich zu umgehen, haben sich die Entwickler von OnLive dazu Gedanken gemacht und erschaffen ein System, bei der die Hardware des Benutzers

²⁰ <http://www.onlive.com>

Hauptteil

absolut keine Rolle mehr spielt, sofern mindestens eine 1,5 bis 5 Mbit/s Internetverbindung vorliegt.

Und das funktioniert wie folgt: sämtliche Spiele werden auf Hochleistungs-Servern installiert und ausgeführt. Der Spieler kann dann jederzeit bequem drauf zugreifen sofern er sich das Spiel „gekauft“ hat und in seinem OnLive-Account freigeschaltet ist. Auf den Servern werden dann die rechenlastigen 3D-Berechnungen und die Kompression der Bilddaten durchgeführt. Die Steuerbefehle des Benutzers werden dann an den Server weitergeleitet, verarbeitet und als eine Art Videostream zum Spieler zurück geschickt. Das Entpacken der komprimierten Bilddateien erfolgt über den Browser.

Die Softwareumgebung entfällt auch hier, denn alles was OnLive anbietet und streamed wird komplett im Browser nutzbar sein und ist weder an einem PC, Mac oder anderer Plattformen / Software gebunden. Im Vergleich zu Quake Live braucht man hier keine Software vorher runterzuladen. Mit einer 1,5 Mbit/s laufen die Spiele in einer guten Auflösung, wer jedoch das HD Erlebnis genießen will, muss mindestens eine Verbindungsgeschwindigkeit von 5 Mbit/s vorweisen um alles ruckelfrei spielen zu können.

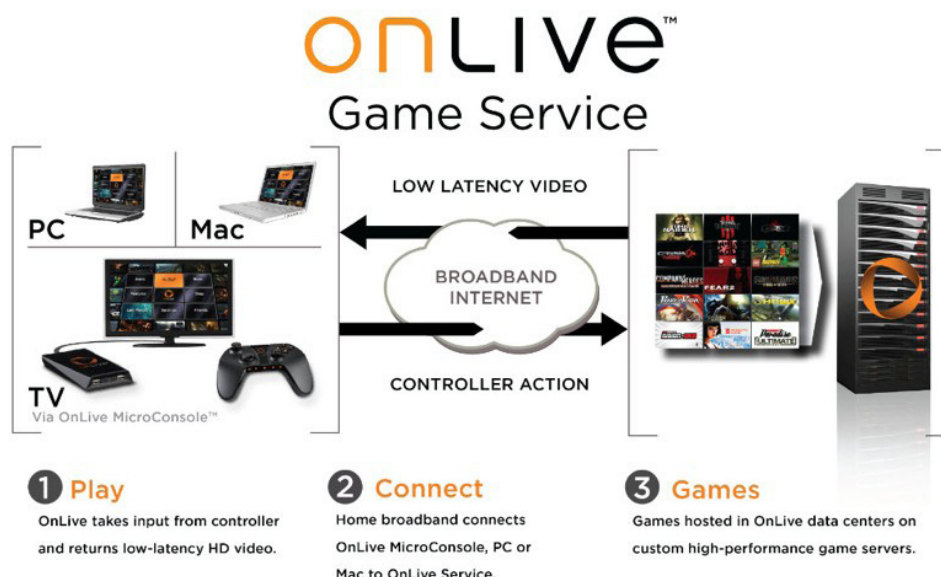


Abb. 13 - OnLive - How it works

Hauptteil

Neben OnLive gibt es derzeit noch drei weitere Konkurrenten, die mit diesem Streamingverfahren auf gute Aussichten hoffen.

- GaiKai will PC und Konsolenspiele über ein Flash-Plugin im Browser streamen.
- Otoy plant in Zusammenarbeit mit AMD hochauflösende Games für das Handy und mobile Geräte zu liefern.
- PlayCast plant das streamen von Spielen über das israelische Kabelfernsehnnetz.

OnLive ist noch eine relativ junge Plattform (Release 17. Juni 2010), und auch wenn der Ansturm zum vorherigen Betatest in den USA ziemlich groß war, kann man trotzdem noch keine Prognosen abgeben ob sich solche Onlinedienste in der breiten Masse durchsetzen werden. Das Plattformen wie OnLive & Co. jedoch nicht so schnell scheitern werden, liegt mitunter an der Spieleindustrie, die das Potential „Kostenersparnis“ erkannt haben.

Vorteile des Streamens von kompletten Spielen:

- Der Spieler ist nicht von der Hardware abhängig und kann auch mit kostengünstigen Rechnern, hochauflösende Grafikwunder problemlos spielen
- Das Problem des „cheatens“ was gerade im PC Bereich weit verbreitet ist, lässt sich damit beheben
- Keine lästiges Runterladen mehr von Updates und Patches. Alles wird automatisch auf den Servern erledigt
- gekaufte Spiele sind jederzeit online verfügbar, d.h. Verlust von Beschädigung der CD's oder DVD's spielt keine Rolle mehr

Hauptteil

- Transport-, Verpackungs- und Herstellungskosten würden sich bei den Herausgebern der Spiele deutlich senken und Gewinne optimieren
- wichtigster Punkt für die Spieleindustrie wäre das eingrenzen von Raubkopien und der Handel mit Gebrautspielen

Nachteile des Streamens:

- Man ist an einem monatlichen Abo gebunden, auch wenn das auf lange Sicht gesehen eventuell die Hardwareanschaffungskosten unterbietet
- Keine Internetverbindung = keine Spiele
- Kein Provider kann garantieren das die Leitungen konstant 5 Mbit/s hergeben.
- Mögliche Verbindungsabbrüche oder Serverprobleme von OnLive & Co. verhindern den Spielspaß

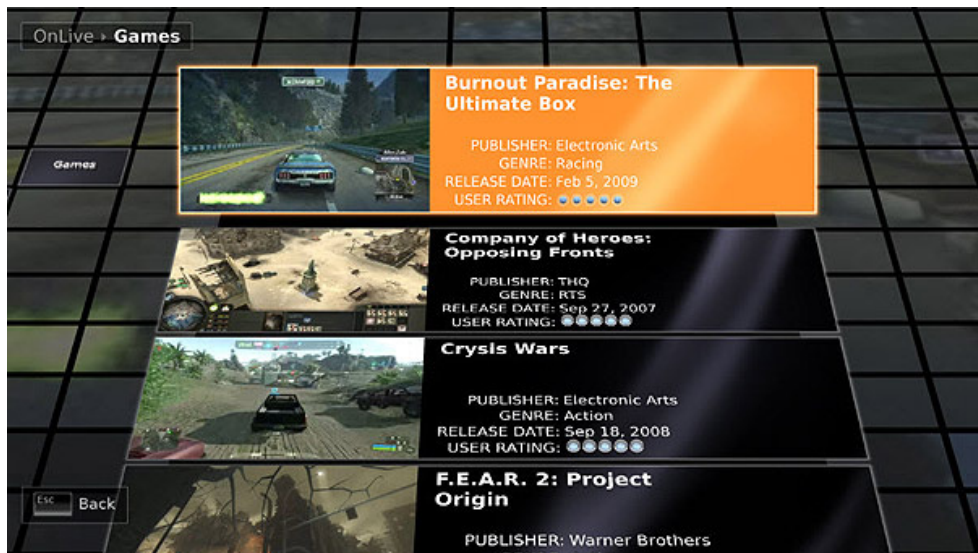


Abb. 14 - OnLive Auswahlmenü von mehreren Spieltitel

Hauptteil

Eine weitere Möglichkeit wäre, dass solche Streaming-Plattformen die Spielkonsole völlig verdrängen werden. Denn das was PlayCast derzeit dabei ist umzusetzen, macht die extra Anschaffung einer Konsole wie z.B. Wii, Xbox oder Playstation völlig überflüssig. Eine kleine Box, an der Controller und Headset angeschlossen werden können, macht das Ganze nicht nur am PC sondern auch am TV nutzbar. Die Konsolenhersteller Microsoft, Sony und Nintendo sehen das Ganze noch relativ gelassen.

„Ich glaube nicht, dass die Technologie es mit dem aufnehmen kann, was wir lokal auf unseren Maschinen bieten“, sagt Jerry Johnson, Europe General Manager von Xbox Live²¹.

Sollten sich jedoch der bisher vier Anbieter solcher Plattformen durchsetzen, gibt es keine Anzeichen die dagegensprechen die Konsolen in die Besenkammer zuverfrachten.

2.3.6. Cloud Computing

Am Anfang des Kapitels trat der Begriff „Cloud Computing“ auf. Um das ganze System von Cloud Computing zu erklären, bedarf es wohl einer extra Arbeit und würde hier zu weit ausschweifen. Aber einen kurzen Blick reinzuwerfen ist unabdinglich, da Google's Betriebssystem Chrom OS, im nächsten Beispiel, gänzlich darauf basiert.

The Cloud, die Wolke, gilt als kommende technische Revolution im Internet. Dokumente, Internetseiten, Videos, Fotos und Software soll zukünftig nicht mehr auf den eigenen Rechnern abgelegt werden, sondern „irgendwo in der Wolke“. Die Wolke besteht aus riesigen, überall auf der Welt verteilten, Datenzentren. Wo genau dann die eigenen Daten liegen spielt keine Rolle mehr, denn sie sind jeder Zeit übers Internet verfügbar, in der „Wolke“ eben.

Gravierende Konsequenzen hat das auch für die Hard- und Softwareherstellen, denn Privatpersonen oder Unternehmen brauchen keine treure Hardware mehr, sondern mieten sich

²¹ http://www.looki.de/forum/diskussionen_zu_news_artikeln_videos_f19/online_xbox_live_boss_sieht_keine_gefahr_durch_cloud_gaming_404140.html

Hauptteil

die entsprechenden Kapazitäten / Ressourcen bei Bedarf in der „Wolke“, um ihre Programme oder Dokumente zu speichern. Das Kaufen von Lizenzen für Programme würde entfallen, weil die Hersteller ihre Software als Service anbieten (z.B. Salesforce in den USA). Google ist ebenso ein Vorreiter und hat bisher den größten Schritt Richtung Cloud Computing getan. Texte, Tabellen, Fotos, der Terminkalender und E-Mails lassen sich über Google erstellen, verwalten und speichern, die nur übers Internet abrufbar sind.

Für den Internetbenutzer wird sich ebenfalls eine Menge ändern. Kosten für die Anschaffung von Programmen entfallen. Es wird nur noch für die jeweilige Dauer oder Häufigkeit der Software bezahlt. Der Internetanschluss ist das einzig notwendige, der die Verbindung zu mehreren hunderttausend Netzwerkrechnern, die eine (Google-)Wolke herstellen.

Vergleicht man das Prinzip mit Soft- oder Hardware-on-Demand, so sieht man einige Parallelen zu OnLive. Spiele und Hardware sind individuell für den Benutzer abrufbar.

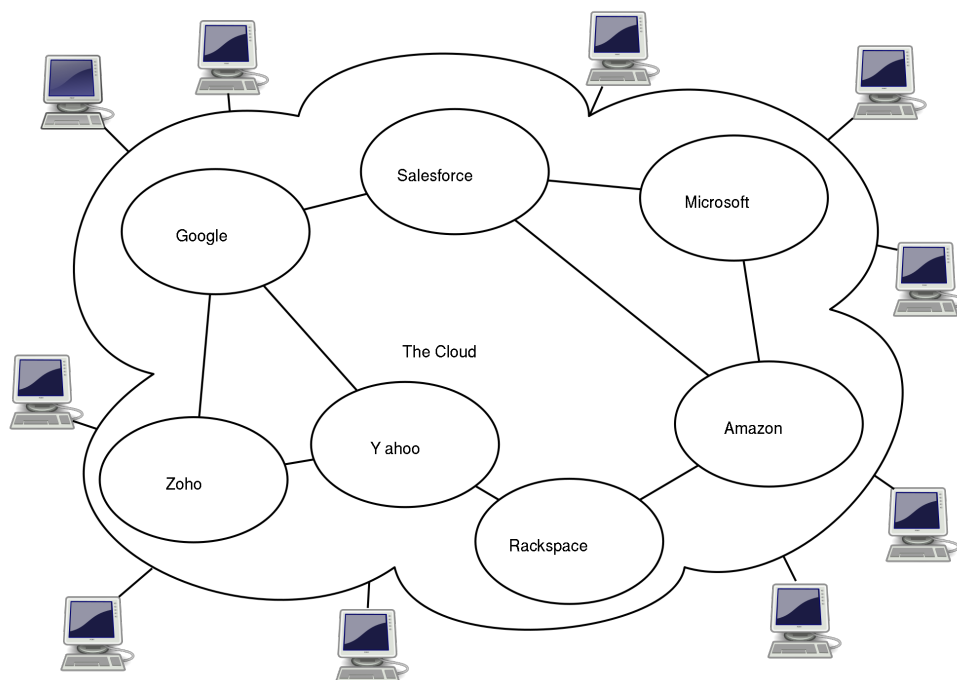


Abb. 15 - Darstellung des Cloud Computing Prinzips / IT-Struktur

2.3.7 Google Chrome OS

Wenn man drüber nachdenkt, womit man sich die meiste Zeit am PC vertreibt - ist es wahrscheinlich das Internet. E-Mails schreiben, Chatten, Nachrichten lesen, wissenswertes nachschlagen, spielen, einkaufen oder Videos gucken beinhalten geschätzte ca. 90% der gesamten Zeit vor dem Rechner. Damit wird der Webbrowser also mit zu den wichtigsten Programm auf dem PC. Das war der Gedanke von Google's Mitarbeiter, als sie den Browser Chrome entwickelt haben. Webapplikationen dynamischer anzeigen, sicheres surfen und vor allem Geschwindigkeit. Chrome ist derzeit der wohl schnellste Browser des es auf dem Markt gibt, und das ohne Konkurrenz²². Ein Grund dafür, ist das er sich mehr Arbeitsspeicher vom System abzweigt und jeder offene Tab als eigene Anwendung behandelt (sieht man eindeutig im Taskmanager). Um Google's Idee weiter fortzusetzen muss man noch einen Schritt weiter gehen.

Wie lange dauert es bis ich endlich im Internet bin? Denn das Internet wäre eventuell der einzige Grund warum ich den Computer hochfahre. Die schnellsten Rechner brauchen ca. 45 Sekunden zum booten bis Windows alle Anwendungen geladen hat und man endlich den Browser benutzen kann um online zu gehen. Google's Entwickler war das eindeutig zu lange und haben deswegen ein neues Konzept aufgestellt.

Wieso sollte nicht der Browser als neues Betriebssystem fungieren? Wofür Windows hochfahren, wenn wir eh alles im Browser online erledigen können. Keine Software mehr verwalten, keine nervigen Updates installieren, keine Systemabstürze (Blue-screen), verloren gegangene Daten oder verwirrende Systemeinstellungen gehören damit zur Vergangenheit. Es ist einfach nur noch der Browser da. Und genau das bietet „Google Chrome Operating System“ - Chrome OS.

Möglich gemacht wird das Ganze durch das Prinzip des Cloud Computing. Dokumente, Musik hören, Chatten etc. wird alles im Browser verfügbar sein und absolut nichts davon wird auf dem heimischen Rechner gespeichert. Denn alles ist irgendwo in der

²² <http://www.welt.de/wirtschaft/webwelt/article2388931/Googles-Internet-Browser-Chrome-ist-eine-Rakete.html>

Hauptteil

Wolke verfügbar. Google arbeitet derzeit eifrig mit namenswerten Firmen zusammen um das Google Netbook zuzubauen. Das Betriebssystem soll für speziell Netbooks oder ähnlich schwache Rechner konzipiert worden sein. Die Veröffentlichung des Ganzen soll bis zum Ende des Jahres erfolgen. Laut eigenen Angaben soll der Rechner innerhalb von sieben Sekunden hochgefahren und betriebsbereit sein.

Eine direkte Konkurrenz zu Windows PC's oder Mac wird das Netbook nicht sein. Denn erstens wird das Netbook nur seine kompletten Funktionalitäten auffahren, wenn das Internet vorhanden ist. Und zweitens werden bisher zwar viele, aber noch nicht alle Webapplikationen verfügbar sein, die man normalerweise auf einen üblichen PC installieren kann. Ebenso hardwarelastige Programme wie Photoshop oder Premiere & Co. sind nicht nicht die „Zielgruppe“ auf die es Google abgesehen hat, somit wird hier keine direkte Verdrängung von PC's oder Macs auf dem Markt erwartet.

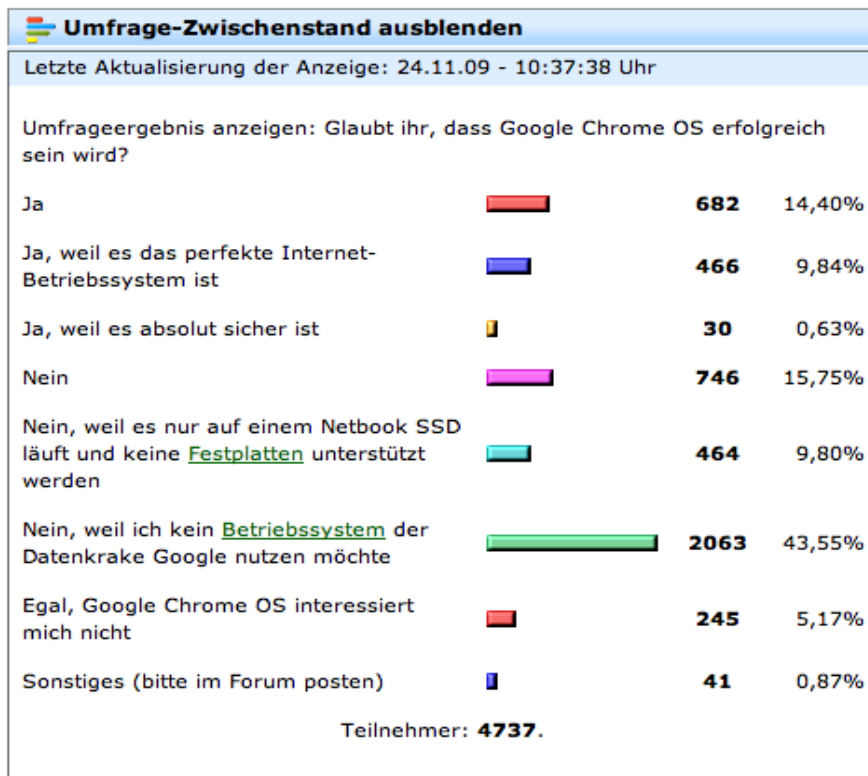


Abb. 16 - Umfrage zu dem Betriebssystem Chrome OS von chip.de

Hauptteil

Doch was erstmal alles ziemlich cool klingt, bringt auch eine Menge Kritik und Unsicherheiten hervor. In einer Umfrage von Chip.de (Abb. 16) hält sie die Begeisterung von Chrome OS in Grenzen. Hervorstechender Grund ist der Zweifel an der Sicherheit der eigenen Daten. Weitere Gründe:

- Die Benutzer begeben sich sprichwörtlich in eine Art „Glashaus“, denn keine der Daten sind lokal gespeichert
- Die Monopolstellung von Google auf diesem Gebiet, sort für einige für Bedenken was mit den Daten eventuell an Mißbrauch geschehen könnte

Google arbeitet jedoch heftig derzeit an der Sicherheit ihres Systems bzw. der Wolke, um die Daten der Benutzer vor allen möglichen Angriffen von Außen zu schützen.

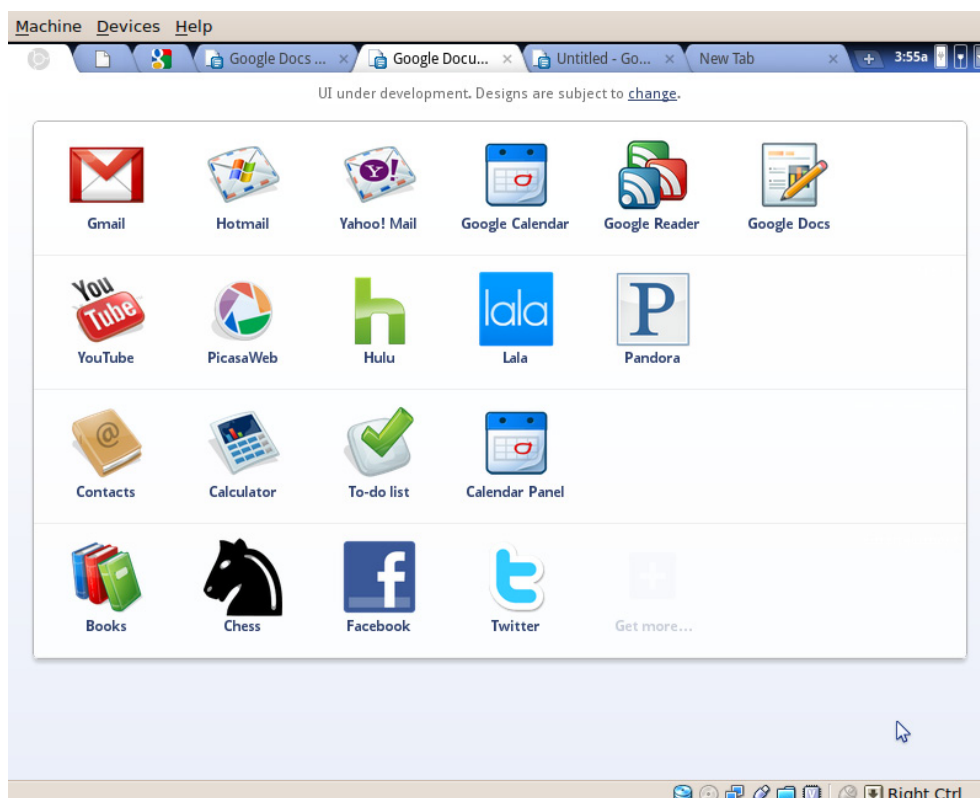


Abb. 17 - Mögliches Design von Chrome OS

Einmal bitte updaten...

Damit möglichst eine breite Masse die Umstellung auf HTML5, CSS3 und Co. mitkriegen und nutzen können ist es Voraussetzung einen relativ aktuellen Browser zu benutzen. Denn um alle neuen Features und Webapplikationen voll nutzen zu können, muss der Browser alle Elemente auslesen können. Und damit kommen wir zum größten Problem, das auf die meisten Internetbenutzer weltweit zutrifft - sie benutzen einen veralteten Browser.

Sieht man sich aktuelle Statistiken und Marktanteile der verschiedenen Browserhersteller an, so sieht es in Deutschland noch relativ ernüchtert aus. Ziemlich genau die Hälfte benutzt einen Firefox als Standardbrowser. Das heisst auch, dass man davon ausgehen kann das die Meisten der 50% auch eine aktuelle Version des Browsers benutzen, da man über Mozilla von neuen Updates schnell Wind bekommt und ein Update sich jedesmal lohnt.











1		Firefox		50.0%	-0.2%	50.2%
2		Internet Explorer		32.9%	+0.2%	32.7%
3		Safari		6.9%	-0.1%	7.0%
4		Chrome		5.5%	±0.0%	5.5%
5		Opera		3.5%	+0.1%	3.4%

Abb. 18 - Statistik der Marktanteile des Browsers in Deutschland

Sieht man sich aber mal eine globale Statistik der Marktanteile an, so hat der Internet Explorer die Nase immernoch vorn. Abgesehen davon, dass es bisher immer Probleme mit dem IE in Verbindung mit einer richtigen Darstellung via CSS gab, verfügt der IE bis heute die mit an wenigsten implementierten HTML5 Elemente. Da HTML5 und CSS3 sich immernoch im noch im Entwicklungsstadium befinden, ist es noch zu früh zu sagen ob sich der Browser damit zukunftsorientiert verhält oder nicht. Aber unter den IE Benutzer herrscht eine große Faulheit was das Updaten auf eine neuere Version angeht. So benutzen immernoch ca. 38% aller IE Verfechter die Version 6.0, welche ja als

Hauptteil

großes Sorgenkind im Allgemeinen gilt. Grund dafür sind zum Teil hohe Kosten, die Firmen und Unternehmen belasten würde um einen Wechsel oder Update vorzunehmen. Auf der anderen Seite gibt es dafür aber auch keinen Grund, da immernoch etliche Werbeagenturen die Darstellungen von Webseiten für den IE 6 durch Browserweichen o.ä. anpassen. Erst wenn gewisse Webapplikationen unverzichtbar werden, große Unternehmen den ersten Schritt tun oder sich wirklich ein Großteil der Internetauftritte umsteigen auf HTML5 und CSS3, werden die Anderen Stück für Stück nachrücken.

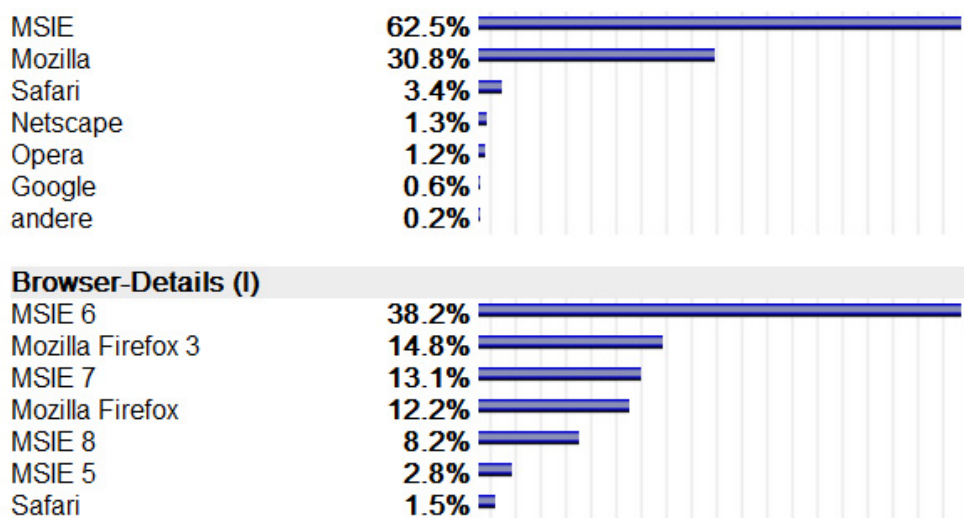


Abb. 19 - Statistik der weltweiten Browser-Marktanteile

Der Verlauf der Marktanteile von Beginn an des Jahres bis Juli 2010, bestätigt diesen Trend, der teilweise durch Google's Browser Chrome zu verzeichnen ist. Während der Internet Explorer und Mozilla Firefox jeweils ca. 2% auf dem Markt einbüßen müssen, steigert sich Chrome um diesen Wert. Der Browser sticht nicht nur durch seine extreme Schnelligkeit hervor sondern, kann was das Implementieren von HTML5 Elementen angeht, mit Firefox mithalten. Safari und Opera verzeichnen ebenfalls einen Zuwach in diesem Jahr, jedoch ist dieser so gering, das man ihn getrost ignorieren kann.

Für alle Personen die also ein Teil dieser sogenannten „Internetrevolution“ sein wollen kann man nur raten - einmal auf

Hauptteil

den großen Update Button drücken, denn ansonsten wird man viele beeindruckende und coole Features verpassen, die in den nächsten Monaten und Jahren folgen.

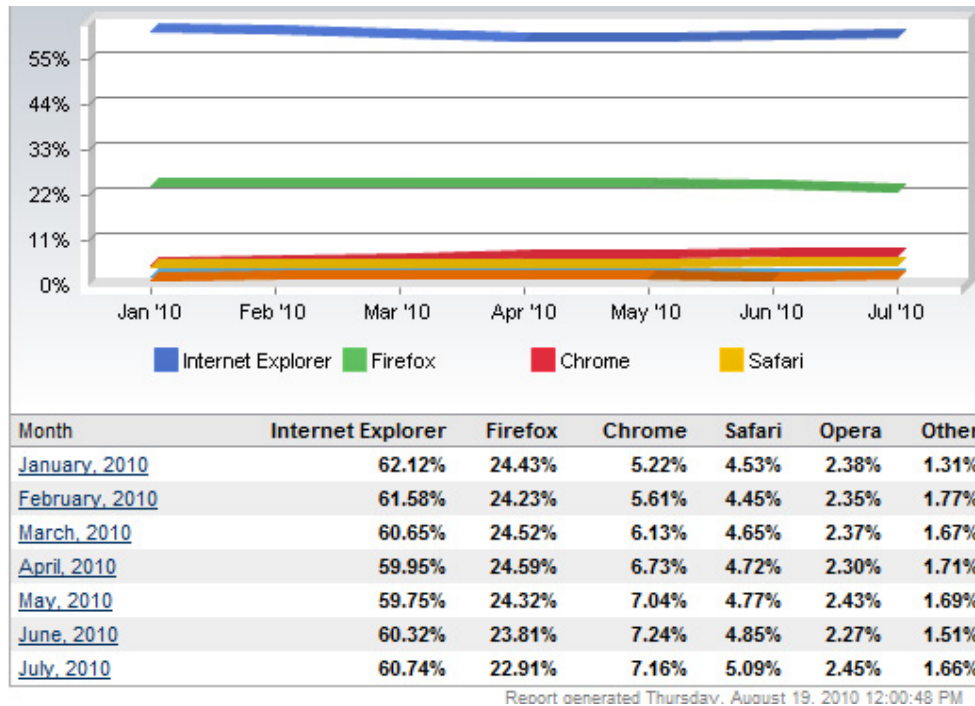


Abb. 20 - Verlauf der Marktanteile von Januar - Juli 2010

Schluss

3.1 Fazit

Kann man nun den Browser als hardwareunabhängiges, eigenständiges Medium betrachten? Ich würde mit einem zögerlichen „ja“ und einem entschlossenden „er ist auf dem besten weg dahin“ beantworten.

Das zögerliche „Ja“, steht für die aktuelle Entwicklung des Browsers. Er hat definitiv das Potential, mehr zu sein als ein schlichtes Programm, was nur Webseiten anzeigt. Aber dennoch dauert so ein Wandel mehrere Jahre, bis er den Zugang zu breiten Masse in seiner neuen Funktion bekommt. Das entschlossene „er ist

Hauptteil

auf dem besten Weg dahin“, zeigt vielversprechende Technologien und Neuansätze. Durch die Streaming- und Cloud Computing Technologie, ist er als eigenes Medium bzw. Betriebssystem, durchaus von der Hardware losgelöst. Ressourcenlastige Anwendungen lassen sich trotz veralteter Hardware anwenden, und falls doch nicht, kann ich sie mir jederzeit in der Wolke mieten.

HTML5, CSS3 und JavaScript machen den Browser wesentlich dynamischer und kompatibler für jede Plattform. Durch die daraus resultierenden Webapplikationen, werden installierte Programme auf dem Rechner überflüssig und erweitern den Browser um viele neue Funktionalitäten.

3.2 Standpunkt in 5 bis 10 Jahren

Eine Prognose abzugeben, für die weitere Entwicklung des Internets, ist wie ein Wetterbericht - ziemlich ungenau und nicht immer zutreffen. Der sprunghafte Wechsel von neuen Technologien, Erfindungen oder Trends machen das Ganze nicht sehr berechenbar. Geht man aber vom heutigen Stand aus und was das Cloud Computing und der Browser als immer stärker werdendes Programm möglich machen, trifft das Motto: „All-in-One“ ziemlich gut zu.

Den optimalen Vergleich biete dazu die Mobile-Branche. Die Entwicklung vom einfachen Handy mit dem man nur telefonieren und SMS schreiben konnte, über PALM Communicator bis hin zu den Alleskönnern - den Smartphones.

Haben Plattformen wie OnLive, die mit den ressourcen- und hardwarelastigsten Anwendungen, eine optimale Lösung bieten um den Anwender von der Hardware abzukoppeln, so kann das im Zusammenhang mit Googles Betriebssystem eine überaus starke Kombination ergeben. Sollten dann auch noch aufwendige Programme wie Photoshop & Co. in der „Wolke“ bzw. in Chrome OS problemlos laufen, könnte es die bisherigen Betriebssysteme (Windows, Linux, Mac OS) wie wir sie derzeit nutzen, komplett ablösen. Vorausgesetzt natürlich das alle Si-

Schluss

cherheitsvorkehrungen getroffen werden um den einzelnen Anwender zu schützen.

Der ständige Zwang zum Einkauf neuer High-End Geräte entfällt, denn alles lässt sich mit auch mit einem mittelmäßig starken Rechner im unteren Preissegment ausführen. Trotz aller Kritik was dieses gesamte Prinzip angeht, ist meine persönliche Meinung, dass gerade Google's Vorhaben eine ziemlich gewagte aber durchaus interessant „Neugestaltung“ des Internets bzw. dessen Nutzen ist.

Müsste ich den Wetterbericht dann noch abmoderieren würde wohl nur eines passen - Heiter mit „wolkigen“ Aussichten.

Literaturverzeichnis

Bücher

Kröner, Peter: HTML5 - Webseiten innovativ und zukunftssicher, München 2010, Open Source Press

Internet

Amberie / Copperlicht

<http://ambiera.com/copperlicht/index.html>

Aufgerufen: 26.07.2010 - 10:51

Aseantic.com / Die Zukunft der Videospiele?

<http://www.aseantic.com/de/Blog/archive/2010/02/11/OnLive-Die-Zukunft-der-Videospiele.aspx>

Stand: 11.02.2010 - 8:30 Aufgerufen: 14.07.2010 - 23:10

Browser-Fuchs.de / Google ändert seine 3D-Web-Strategie

<http://www.browser-fuchs.de/?p=1087>

Aufgerufen: 18.07.2010 - 22:12

Browser-Statistik.de

<http://www.browser-statistik.de/>

Aufgerufen: 18.08.2010 - 19:01

Chip.de / Google Chrome OS: Alle Infos, Bilder

http://www.chip.de/news/Google-Chrome-OS-Alle-Infos-alle-Bilder_39016139.html

Stand: 20.11.2009 - 09:00 Aufgerufen: 05.08.2010 - 13:45

Engadget.com / OnLive killed the game console star?

<http://www.engadget.com/2009/03/24/onlive-killed-the-game-console-star/>

Stand: 24.03.2009 - 02:51 Aufgerufen: 12.08.2010 - 19:13

Faz.net / Internetgiganten kämpfen um die Wolke

http://www.faz.net/s/RubE2C6E0BCC2F04DD787CDC274993E94C1/Doc~E1F5DC58852F24B3F9DC6AA81B0DF3977~ATpl~Ecommon~Scontent.html?rss_googlefeed

Stand: 05.05.2008 Aufgerufen: 18.08.2010 - 20:14

Faz.net / Google Chrome: Große Pläne mit dem kleinen Browser

<http://faz-community.faz.net/blogs/netzkonom/archive/2008/09/02/google-chrome-das-trojanische-pferd.aspx>

Stand: 02.09.2008 - 09:42 Aufgerufen: 11.08.2010 - 16:27

Faz.net / Chrome: Das Betriebssystem der nächsten Generation

<http://faz-community.faz.net/blogs/netzkonom/archive/2009/11/19/chrome-os-das-betriebssystem-der-naechsten-generation.aspx>

Stand: 19.11.2009 - 21:25 Aufgerufen: 11.08.2010 - 15:53

Glgf.org / WebGL for the lazy

<http://www.glgf.org/>

Aufgerufen: 25.07.2010 - 18:38

Golem.de / XML3D - Browser lernen Echtzeit-Raytracing

[http://www.golem.de/showhigh2.php?file=/1003/73645.html&wort\[\]=google&wort\[\]=rendering](http://www.golem.de/showhigh2.php?file=/1003/73645.html&wort[]=google&wort[]=rendering)

Stand: 05.03.2010 - 16:31 Aufgerufen: 01.07.2010 - 11:29

Golem.de / Firefox wird 3D-fähig

<http://www.golem.de/0909/69946.html>

Stand: 20.09.2009 - 12:03 Aufgerufen: 26.07.2010 - 12:13

Golem.de / Google macht Browser 3D-fähig

<http://www.golem.de/0904/66622.html>

Stand: 21.04.2009 - 21:15 Aufgerufen: 01.07.2010 - 13:30

Golem.de / 3D-API fürs Web

<http://www.golem.de/0903/66105.html>

Stand: 25.03.2009 - 10:36 Aufgerufen: 20.07.2010 - 16:47

Golem.de / Quake Live: Indizierter Klassiker hat Startprobleme

<http://www.golem.de/0902/65506.html>

Stand: 25.02.2009 - 12:31 Aufgerufen: 02.07.2010 - 12:28

Google / Google's O3D Plug-In API
<http://code.google.com/intl/de-DE/apis/o3d/>
Aufgerufen: 01.07.2010 - 13:12

Google / Quake II GWT Port
<http://code.google.com/p/quake2-gwt-port/>
Aufgerufen: 02.07.2010 - 15:23

Handelsblatt.com / Chronik der großen "Browserkriege"
<http://www.handelsblatt.com/technologie/it-internet/chronik-der-grossen-browserkriege;2031246;3#bqStart>
Stand: 18.03.2009 - 10:00 Aufgerufen: 26.06.2010 - 21:32

Heise Online / Google Chrome überholt die Konkurrenz
<http://www.heise.de/newsticker/meldung/Google-Chrome-ueberholt-die-Konkurrenz-202963.html>
Stand: 03.09.2008 - 18:40 Aufgerufen: 03.08.2010 - 12:12

Heise Online / Open-Source-Alternative zum Browser-Shooter Quake Live
<http://www.heise.de/newsticker/meldung/Open-Source-Alternative-zum-Browser-Shooter-Quake-Live-7305.html>
Stand: 05.07.2009 - 13:26 Aufgerufen: 02.07.2010 - 16:17

Heise Online / Webkit: 3D-Transformationen mit CSS
<http://www.heise.de/newsticker/meldung/Webkit-3D-Transformationen-mit-CSS-6265.html>
Stand: 17.07.2009 - 15:54 Aufgerufen: 25.07.2010 - 18:23

IGM / Wolkige Aussichten
<http://igmonline.biz/2009/11/wolkige-aussichten/>
Stand: 17.09.2009 Aufgerufen: 18.8.2010 - 21:23

IGN UK / OnLive introduces the future of gaming
<http://uk.pc.ign.com/articles/965/965535p1.html>
Stand: 23.03.2009 Aufgerufen: 12.08.2010 - 19:36

Informatik TU Chemnitz / Geschichte des Browsers
<http://vsr.informatik.tu-chemnitz.de/proseminare/www03/doku/browser/geschichte.htm>
Aufgerufen: 25.06.2010 - 12:34

Informationweek.com / Google Chrome: Browser or Cloud Operating System

http://www.informationweek.com/blog/main/archives/2008/09/google_chrome_b.html

Stand: 02.09.2008 - 08:46 Aufgerufen: 12.08.2010 - 14:21

Netmarketshare.com

<http://marketshare.hitslink.com/browser-market-share.aspx?qprid=0>

Aufgerufen: 18.08.2010 - 22:18

PC Welt / Browser Geschichte: Die wichtigsten Browser der letzten 15 Jahre

http://www.pcwelt.de/start/software_os/online/praxis/2104328/die_wichtigsten_browser_der_letzten_15_jahre/index3.html

Stand: 23.12.2009 - 14:22 Aufgerufen: 25.06.2010 - 12:45

Satine.org / Snow Stack is here

<http://www.satine.org/archives/2009/07/11/snow-stack-is-here/>

Stand: 11.07.2009 - 13:41 Aufgerufen: 27.07.2010 - 09:37

Six Revisions / Six Questions: Eric Meyer on CSS3

<http://sixrevisions.com/interviews/six-questions-eric-meyer-on-css3/>

Stand: 09.12.2009 Aufgerufen: 24.07.2010 - 18:56

Spiegel.de / Internet Explorer weiter im Aufwind

<http://www.spiegel.de/netzwelt/web/0,1518,709769,00.html>

Stand: 03.08.2010 Aufgerufen: 05.08.2010 - 11:10

Suite101.de / Cloud Computing und Cloud IT

http://internettechnik-netzwerktechnik.suite101.de/article.cfm/cloud_computing_und_cloud_it

Stand: 02.10.2009 Aufgerufen: 17.08.2010 - 23:54

Tom's Hardware / Ballern im Browser mit HTML5

<http://www.tomshardware.de/Quake-II-HTML5,news-244179.html>

Stand: 06.04.2010 - 18:40 Aufgerufen: 02.07.2010: 16:45

Tom's Hardware / Quake Live Opening Beta Week

<http://www.tomshardware.com/news/quake-live-beta-arena-id,7069.html>

Stand: 20.02.2009 - 19:50 Aufgerufen: 02.07.2010 - 17:01

TweakPC.de / Flash bald überflüssig? 3D-CSS macht es möglich
<http://www.tweakpc.de/news/16489/flash-bald-ueberfluessig-3d-css-macht-es-moeglich/>

Stand: 15.07.2009 - 18:11 Aufgerufen: 27.07.2010 - 14:21

Webkit.org / 3D Transforms
<http://webkit.org/blog/386/3d-transforms/>

Stand: 16.07.2009 - 12:02 Aufgerufen: 01.08.2010 - 11:09

Webstandards.org / Interview with Ian Hickson, HTML5 editor
<http://www.webstandards.org/2009/05/13/interview-with-ian-hickson-editor-of-the-html-5-specification/>

Stand: 13.05.2009 Aufgerufen: 03.07.2010 - 10:01

Webstandard.Kulando.de
<http://webstandard.kulando.de/post/2010/03/05/css3-transition-tutorial-menu-mit-slide-effekt-im-apple-style>

Aufgerufen: 28.07.2010 - 13:13

W3C / Working Draft CSS3
<http://www.w3.org/TR/css3-3d-transforms/>

Aufgerufen: 25.07.2010 - 17:59

Internet - Webverzeichnisse / Suchmaschinen

Google.de
<http://www.google.de>

Stand: mehrere Suchanfragen während der Bearbeitungszeit: 22.6 - 31.8.2010

Internet - Nachschlagewerke

Wikipedia
<http://www.wikipedia.de>

Stand: mehrere Suchanfragen während der Bearbeitungszeit: 22.6 - 31.8.2010

Erklärung zur Selbstständigen Anfertigung

Hiermit erkläre ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Alle Teile, die wörtlich oder sinngemäß einer Veröffentlichung entstammen, sind als solche kenntlich gemacht. Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Unterschrift